

## KOREAN PATENT ABSTRACT (KR)

### PUBLICATION

(51) IPC Code: G06F 17/00

(11) Publication No.: P2002-0004931

(43) Publication Date: 16 January 2002

(21) Application No.: 10-2001-7004228

(22) Application Date: 02 April 2001

(86) International Application No.: PCT/US1999/23008 (1 October 1999)

(87) International Publication No.: WO 2000/21232 (13 April 2000)

(71) Applicant:

International Business Machines Corporation  
Old Orchard Road, Armonk, NY 10504 US

(72) Inventor:

Gopalakshnan Ponani; US.  
Lucas Bruce D; US.  
Maes Stephane H; US.  
Nahamoo David; US.  
Sedivy Jan; CZ

(54) Title of the Invention:

Conversation Browser and Conversational Systems

### Abstract:

A conversational browsing system 10 comprising a conversational browser 11 having a command and control interface 12 for converting speech commands or multi-modal input from I/O resources 27 into navigation request, a processor 14 for parsing and interpreting a CML (conversational markup language) file, the CML file comprising meta-information representing a conversational user interface for presentation to a user. The system 10 comprises conversational engines 23 for decoding input commands for interpretation by the command and control interface and decoding meta-information provided by the CML processor for generating synthesized audio output. The browser 11 accesses the engine 23 via system calls through a system platform 15. The system includes a communication stack 19 for transmitting the navigation request to a content server and receiving a CML file from the content server based on the navigation request. A conversational transcoder 13 transforms presentation material from one modality to a conversational modality. The transcoder 13 includes a functional transcoder 13a to

transform a page of GUI to a page of CUI (conversational user interface) and a logical transcoder 13b to transform business logic of an application, transaction or site into an acceptable dialog. Conversational transcoding can convert HTML files into CML files that are interpreted by the conversational browser 11.

# (19) 대한민국특허청 (KR) (12) 공개특허공보 (A)

(51) . Int. Cl. 7  
G06F 17/00

(11) 공개번호 특2002 - 0004931  
(43) 공개일자 2002년01월16일

(21) 출원번호	10 - 2001 - 7004228		
(22) 출원일자	2001년04월02일		
번역문 제출일자	2001년04월02일		
(86) 국제출원번호	PCT/US1999/23008	(87) 국제공개번호	WO 2000/21232
(86) 국제출원출원일자	1999년10월01일	(87) 국제공개일자	2000년04월13일

(81) 지정국                      국내특허 : 캐나다, 중국, 이스라엘, 일본, 대한민국, 미국,  
EP 유럽특허: 오스트리아, 벨기에, 스위스, 독일, 덴마크, 스페인, 프랑스, 영국, 그리스, 아  
일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투갈, 스웨덴, 핀란드, 사이프러스,

(30) 우선권주장	60/102,957	1998년10월02일	미국 (US)
	60/117,595	1999년01월27일	미국 (US)

(71) 출원인                      인터내셔널 비지네스 머신즈 코포레이션  
포만 제프리 엘  
미국 10504 뉴욕주 아몬크

(72) 발명자                      고파라크리시난포나니  
미국뉴욕주10598요크타운하이츠라드크리프트라이브3073  
루카스브루스디  
미국뉴욕주10598요크타운하이츠밀폰드로드2408  
마에스스테판에이치  
미국코네티컷주06811덴버리윈터그린힐로드1  
나하무데이빗  
미국뉴욕주10605화이트플레인스엘름우드로드12  
세디비잔  
체코프라하유레자11

(74) 대리인                      허정훈  
김성택

심사청구 : 없음

(54) 대화식 브라우저 및 대화식 시스템

요약

대화식 브라우저 시스템(10)은 I/O 자원(27)으로부터의 음성 명령 또는 다중 양식 입력을 네비게이션 요구로 변환하는 명령 및 제어 인터페이스(12)와, CML 파일이 사용자에게 표시하기 위한 대화식 사용자 인터페이스를 나타내는 메타 정보를 포함하는 CML(대화식 마크업 언어) 파일을 분석 및 해석하는 CML 프로세서(14)를 구비한 대화식 브라우저(11)를 포함한다. 이 시스템(10)은 명령 및 제어 인터페이스에 의해 해석하기 위한 입력 명령을 디코딩하고, 합성 오디오 출력을 발생하기 위해 상기 CML 프로세서에 의해 제공되는 메타 정보를 디코딩하는 대화식 엔진(23)을 포함한다. 브라우저(11)는 시스템 플랫폼(15)을 통하여 시스템 호출을 통해 대화식 엔진(23)에 액세스한다. 상기 시스템은 네비게이션 요구를 내용 서버로 송신하고, 그 네비게이션 요구에 기초하여 상기 내용 서버로부터 CML 파일을 수신하는 통신 스택(19)을 포함한다. 대화식 트랜스코더(13)는 표시 자료를 하나의 양식으로부터 대화 양식으로 변환한다. 상기 대화식 트랜스코더(13)는 GUI의 페이지를 CUI(대화식 사용자 인터페이스)의 페이지로 변환하는 기능 트랜스코더(13a) 및 어플리케이션, 트랜잭션 또는 사이트의 비즈니스 논리를 허용 가능한 다이얼로그로 변환하는 논리 트랜스코더(13b)를 포함한다. 대화식 트랜스코더는 HTML 파일을 상기 대화식 브라우저(11)에 의해 해석되는 CML 파일로 변환할 수 있다.

대표도  
도 1

명세서

기술분야

이 출원은 1998년 10월 2일자로 출원된 가출원 미국 출원 번호 제60/102,957호 및 1999년 1월 27일자로 출원된 미국 출원 번호 제60/117,595호를 기초로 한다.

본 발명은 정보에 액세스하는 시스템 및 방법에 관한 것으로, 특히 현재의 네트워킹 기간 시설(networking infrastructure)을 변경할 필요 없이 트랜잭션 및 정보 서비스의 동일한 बैं크에 액세스하도록 순수 GUI(그래픽 사용자 인터페이스) 양식 및 순수 음성 인터페이스 양식이 개별적으로(또는 조합하여) 사용될 수 있게 하는 표준 네트워크 프로토콜(HTTP 등)에 다양한 정보 공급원으로서의 액세스의 통일을 제공하는 대화식 브라우저에 관한 것이다.

배경기술

현재, 정보 및 트랜잭션에 전화 액세스하는 IVR(대화식 음성 응답) 서비스를 광범위하게 사용하고 있다. IVR 시스템은 구두 지정 다이얼로그를 사용하고, 통상적으로 이하와 같이 동작한다. 사용자는 IVR 시스템을 다이얼하고 소정 메뉴 및 특정 정보에 액세스하기 위한 선택을 제공하는 오디오 프롬프트를 청취한다. 각 선택은 전화 키패드 상의 하나의 번호로 지정되거나 사용자에게 의해(음성 인에이블 IVR로) 발음된 워드와 관련되고, 사용자는 적절한 버튼을 누르거나 적절한 워드를 발음함으로써 원하는 선택을 행한다. 종래의 IVR 어플리케이션은 일반적으로 제조자에 의해 상이한 HW(하드웨어) 플랫폼에 대해 다양한 구현으로 제공되는 특수한 스크립트 언어로 기록되어 있다. 그러한 IVR 어플리케이션의 개발 및 유지 보수는 제한된 스태프(staff)를 요구한다. 종래의 IVR 어플리케이션은 특수한(및 고가의) 전화 HW(하드웨어)를 사용하고, 각 IVR 어플리케이션은 유산 데이터베이스 서버에 액세스하기 위해 상이한 SW(소프트웨어) 계층을 사용한다. 이들 계층은 각 어플리케이션에 대해 특수하게 설계되어야 한다.

더욱이, IVR 시스템은 GUI 또는 DTMF 및 음성 이외의 다른 양식을 취급하도록 설계되지는 않는다. 종래의 아날로그 접속에서 이진 데이터 및 음성을 혼합하는 것이 가능하지만, 종래의 IVR을 수신기로서 이용하여 이진 데이터 및 음성을 혼합하는 것은 불가능하다. 따라서, IVR 시스템은 일반적으로 데이터/이진 입력 및 음성이 병합되지 않게 한다. 현재, 그러한 서비스는 이진 접속(예컨대, 모뎀의 형태)을 취급하도록 구성된 별개의 시스템을 필요로 한다. 가까운 장래에, 음성 IP(VoIP) 및 무선 통신(예컨대, GSM)은 음성 및 데이터의 동시 송신을 가능하게 한다. 현재, 하나 이상의 동시

호출이 이진 및 음성의 동시 교환을 위해 필요하거나(특수한 작업을 적절하게 처리하는데 유용하게 되는 것이 이후 설명되므로), 데이터의 비동기 송신을 위한 이후의 호출 또는 회신을 필요로 한다. 이것은 일반적으로 불편하다. 특히, 데이터 교환은 압축 음성 및 음성 UI를 생성하는 것과 관련된 정보를 송신 또는 수신하는 것보다 많을 수 있고, 또한 UI(예컨대, GUI)에 양식을 추가하는 데 필요한 정보일 수 있다. 서비스가 예컨대, 질문의 결과 및 표시 자료가 또한 GUI 자료(예컨대, 넷스케이프 네비게이터와 같은 GUI 브라우저 상에 표시되는 HTML)를 결과로 생성하는 음성 입력/웹 출력(또는 음성 입력/웹 및 음성 출력)을 제공하도록 다중 라인을 사용한다고 가정하면, 서비스 제공자는 모든 IT 기간 시설 및 백엔드를 적절하게 네트워킹되도록 부가해야 하고 자신의 백엔드, IVR 및 웹 서버를 동기화시켜야 한다. 간단하지만 매우 어려운 작업이 표시의 GUI 또는 HTML 부분에 대하여 음성 표시 자료의 동작/전개 사이의 조정이다.

이동 및 홈 컴퓨팅의 전개뿐만 아니라 인터넷의 보급이 급속히 증가함에 따라, 네트워킹된 PC, NC, 정보 키오스크(kiosk) 및 정보 서비스와 트랜잭션에 액세스하기 위한 다른 가전 장치(IVR 전화 서비스와 대비되는 것으로서)의 사용도 광범위하게 확산되고 있다. 실제로, 인터넷 및 인트라넷의 폭발적인 증가가 데이터의 변환 및 트랜잭션이 프록시, 서버 및/또는 트랜스코더를 통해 실행되는 GUI "웹 브라우저"로서 공지되어 있는 것을 통해 액세스 가능한 모든 가능한 정보 소스, 데이터베이스 또는 트랜잭션에 가상적인 액세스를 제공하고 있다.

통상적으로, 웹 브라우저는 사용자가 WWW(월드 와이드 웹 또는 "웹")(즉, 인터넷 상에서 HTTP 서버의 요구를 생성하도록 HTTP(하이퍼텍스트 전송 프로토콜)를 이용하는 클라이언트 프로그램) 상의 정보를 시청 및 이 정보와 대화할 수 있게 하는 어플리케이션 프로그램(또는 클라이언트 프로그램)이다. 인터넷 상의 HTTP 서버는 표준 HTML(하이퍼텍스트 마크업 언어)로 기록되는 "웹 페이지"를 포함한다. 인터넷 웹 페이지는 HTTP 서버로부터 패킷 교환 네트워크를 통해 액세스될 수 있고, 웹 브라우저에 의해 해석된 후 그래픽 형태로 사용자에게 표시된다. 사용자에게 표시되는 텍스트 정보는 정보의 새로운 소스로의 강조된 하이퍼링크를 포함한다. 사용자는 예컨대, 웹 브라우저에 의해 표시하기 위한 새로운 웹 페이지를 다운로드하기 위해 마우스로 클릭함으로써 하이퍼링크를 선택할 수 있다. 인터넷을 통한 유산 데이터베이스로의 액세스는 라이브와이어(LiveWire) 및 JDBC(자바 데이터베이스 상호 통신 능력) 등의 여러 가지 공지된 표준에 의해 인에이블된다. 더욱이, 웹 페이지는 서버로부터 다운로드되어 브라우저 상에서 또는 브라우저가 상부에 생성되는 시스템의 JVM(자바 가상 머신) 상에서 실행될 수 있는 애플릿(예컨대, 자바 프로그램)과 같은 실행 가능 코드를 포함할 수 있다. 다른 정보는 서버 상에서 실행하고 접속된 브라우저에서의 변화를 추구하는 서브릿(예컨대, 자바 프로그램)에 의해 제공될 수 있다. 애플릿 및 서브릿은 웹 서버 및 어플리케이션이 서로 통신할 수 있게 하는 CGI(공통 게이트웨이 인터페이스) 기능을 포함할 수 있다. 또한, 다른 정보 액세스 방법은 브라우저 상에서 해석 및 실행되는 소정의 프로그램 언어인 스크립트를 포함한다. 이것은 예를 들어, 자바스크립트 및 DHTML(동적 HTML) 언어를 포함한다. 플러그인(plugin)은 브라우저에 의해 다운로드될 수 있고 로컬 장치 상에서 원어를 실행하도록 브라우저에 의해 자동적으로 인식될 수 있으며 브라우저에 의해 (다운로드를 통해) 연속해서 제공되는 인수(argument) 상에서 실행될 수 있는 브라우저 외부의 프로그램이다. CGI 스크립트는 비즈니스 로직을 수행하고 후속 표시 자료를 출력으로서 생성하는 서버측 스크립트이다. 애플릿 및 플러그인은 RMI(원격 방법 실시), 소켓 접속, RPC(원격 절차 호출) 등을 통해 통신할 수 있다. 또한, 복잡한 트랜스코딩안, XML(확장 가능 마크업 언어) 확장 및 스크립팅 언어가 특정 정보 또는 서비스용으로 또는 대화를 간략하게 하는데 사용된다.

전술한 바와 같이, 인터넷 웹 브라우저 및 IVR의 목적은 정보에 액세스하는 것이다. 아래의 예는 전화로 IVR을 통해서, 또한 PC 및 웹 브라우저를 사용하여 인터넷을 통해 정보에 액세스하는 데 사용되는 패러다임이 유사한 것을 나타내도록 बैंकिंग 어플리케이션과 관련하여 일반적인 시나리오를 설명한다. 예를 들어, 일반적인 बैंकिंग ATM 트랜잭션은 고객이 전화를 통해 IVR을 사용하여 저축, 수표 및 신용 카드 계정 사이의 현금 이체를 실행할 수 있게 하여 계정 잔액을 체크할 수 있다. 이들 트랜잭션은 또한 인터넷 액세스를 위한 PC 및 웹 브라우저를 사용하여 실행될 수도 있다. 통상적으로, PC를 사용하여, 고객은 텍스트 메뉴의 형태로 정보를 얻을 수 있다. 전화의 경우에는, 정보는 오디오 메뉴를 통해 표시된다. PC 어플리케이션에서의 마우스 클릭은 전화 버튼을 누르거나 명령을 말하는 것으로 변환된다. 특히, 일반적인 홈 बैंकिंग IVR 어플리케이션은 환영 메시지로 시작한다. 이와 유사하게, 은행의 인터넷 홈 페이지는 화상 및 환영 텍스트를

표시할 수 있고, 사용자가 아래의 예와 같은 서비스의 목록으로부터 선택할 수 있게 한다:

- a. 즉시 계정 정보;
- b. 이체 및 현금 지불;
- c. 펀드 정보;
- d. 수표 정보;
- e. 증권 시세; 및
- f. 도움말.

IVR 어플리케이션을 이용하면, 상기 메뉴는 전화를 통해 사용자에게 의해 실행될 수 있고, 그것에 의해 메뉴 메시지에는 숫자 또는 버튼이 후속하여 사용자가 원하는 옵션을 선택하도록 눌러야 한다:

- a. " 즉시 계정 정보를 원하면, 1을 누르시오" ;
- b. " 이체 및 현금 지불을 원하면, 2를 누르시오" ;
- c. " 펀드 정보를 원하면, 3을 누르시오" ;
- d. " 수표 정보를 원하면, 4를 누르시오" ;
- e. " 증권 시세를 원하면, 5를 누르시오" ;
- f. " 도움말을 원하면, 7을 누르시오" .

IVR 시스템은 DTMF 키 대신에 또는 부가하여 음성 인식을 수행할 수 있다. 사용자가 신용 카드 관련 정보를 얻기를 원한다고 가정하자. 인터넷 기반 어플리케이션을 통해 이러한 정보를 얻기 위해, 사용자는 후속 페이지를 표시하기 위해 메뉴에서 특정 하이퍼텍스트 링크를 클릭한다. 전화 어플리케이션에서는, 사용자는 대응하는 DTMF 신호를 송신하도록 적절한 전화 키를 누른다. 그러면, 재생되는 후속 메뉴는:

- a. " 사용 가능 신용을 원하면, 1을 누르시오" ;
- b. " 미결제 잔액을 원하면, 2를 누르시오" ;
- c. " 당신의 계정이 당좌 계정에 연결되어 있으면, 당신의 신용 카드 잔액을 지불할 수 있는데, 이 경우 3을 누르시오" 일 수 있다.

다시, 사용자는 적절한 키를 누름으로써 원하는 선택을 행할 수 있다.

계속하기 위해, 사용자는 인증 정보를 제공하도록 촉구될 수 있다. 이러한 목적으로, 인터넷 어플리케이션은 예컨대, 사용자의 계정 번호에 대하여 비어 있는 필드 및 사용자의 사회 보장 번호에 대한 나머지를 갖는 메뉴를 표시할 수 있다. 정보가 채워진 후에, 서버에 발송되어, 처리되고, 재생이 포맷되어 사용자에게 다시 전송된다. 전화를 통한 시나리오도 동일하다. IVR 시스템은 (전화를 통해) 사용자에게 자신의 계정 번호를 (DTMF 또는 음성을 통해) 입력하도록 요구하는 오디오 프롬프트를 재생할 수 있고, 정보는 DTMF 신호를 처리하거나 음성을 인식함으로써 사용자로부터 수신된다.

사용자는 자신의 SSN을 입력하도록 촉구될 수 있고, 응답이 유사한 방식으로 처리된다. 처리가 완료될 때, 정보는 서버로 전송되며, 여기에서 계정 정보가 액세스되어 오디오 재생으로 포맷된 후 전화를 통해 사용자에게 재생된다.

전술한 바와 같이, IVR은 웹 브라우저로서 정보 액세스를 위해 동일한 패러다임을 사용하여 동일한 기능을 실행시킨다. 실제로, 자체의 인터페이스 및 양식 차이를 초월하여, IVR 시스템 및 웹 브라우저가 현재 기본적으로 상이한 시스템으로서 설계 및 개발되고 있다. 그러나, 가까운 장래에, 은행 및 대형 회사가 자체의 공용으로 액세스 가능한 정보 소스를 구 IVR을 유지하면서 인터넷으로 이동하게 될 것이다. 불행하게도, 이것은 이들 기구가 동일한 유형의 정보 및 트랜잭션 서비스용의 별개의 시스템을 유지하는 것을 필요로 한다. 은행 및 회사가 기존의 기간 시설을 사용하여 인터넷 상으로 IVR을 통해 정보 및 서비스를 제공할 수 있게 하는 것이 유리하다. 이러한 관점으로, 사용자가 정보에 액세스할 수 있게 하고 IVR 및 종래의 브라우저를 사용하여 인터넷 상에서 트랜잭션을 실행하는 범용 시스템 및 방법이 필요하다.

#### 발명의 상세한 설명

본 발명은 액세스의 모드에 무관하게 표준 프로토콜로 어플리케이션으로의 액세스를 통일시키는 시스템 및 방법에 관한 것이다. 특히, 본 발명은 정보에 액세스하고, 예컨대, TCP/IP, HTTP(하이퍼텍스트 전송 프로토콜) 또는 WAP(무선 어플리케이션 프로토콜)에 기초하는 표준 네트워킹 프로토콜 및 예컨대, HTTP 서버로부터 인터넷 상의 정보를 액세스하기 위한 구조를 이용하여 트랜잭션을 실행하는 범용 방법 및 시스템을 제공하므로, 순수 GUI(그래픽 사용자 인터페이스) 양식 및 순수 음성 인터페이스 양식이 현재 사용 가능한 네트워크의 현재의 기간 시설의 변형을 요구하지 않고 트랜잭션 및 정보 서비스의 동일한 बैं크에 액세스하도록 개별적으로(또는 조합하여) 사용될 수 있다.

본 발명의 일 실시예에서, 종래의 브라우저가 제공되어 전화를 통한 명령을 HTTP 프로토콜로 해석한다. 대화식 브라우저의 도입은 인터넷 및 전화(IVR)를 통합할 수 있게 하고, 그것에 의해 비용을 감소시키며, 그러한 어플리케이션의 커버리지 및 신축성을 증가시킨다. 특히, IVR 어플리케이션을 위해, 대화식 브라우저 또는 (전화 브라우저)는 DTMF 신호 및/또는 사용자로부터의 구두 명령을 해석할 수 있고, 적절한 HTTP 서버로부터의 정보에 액세스하도록 HTTP 요구를 발생하며, 이어서 HTML 기반 정보를 해석하여 오디오 메시지를 통해 사용자에게 제공한다. 대화식 브라우저는 또한 HTTP 서버로부터 HTTP 프로토콜로 수신되는 압축 오디오를 디코드할 수 있고, 그것을 사용자에게 재구성하여 재생할 수 있다. 역으로, 대화식 브라우저는 오디오를 포착하여 분배 인식 및 처리를 위해 서버로 (압축되거나 압축되지 않은) 오디오를 서버로 송신할 수 있다. 오디오가 국부적으로 포착되어 서버로 전송될 때, 이것은 플러그 인(원시 수행)을 이용하거나 또는 예컨대, 사용자의 입력을 포착하도록 오디오 및 멀티미디어 API를 사용하여 자바 애플릿 또는 자바 프로그램을 이용하여 행해질 수 있다.

신규 제안 IVR 구조 및 대화식 브라우저에 있어서, 내용 페이지는 GUI 브라우저와 같은 종래의 모드에 의해 액세스되는 동일한 HTTP 서버 상에 있고, 유산 데이터베이스 액세스 SW 계층 등을 분배하는 동일한 정보 액세스 방법을 사용한다. 즉, IVR은 대화식 브라우저를 갖는 HTTP 서버의 특수한 경우이다. 종래의 GUI 브라우저 및 PC와 유사하게, 대화식 브라우저, 정보 및 질문이 동일한 프로토콜(HTTP)을 사용하여 교환 패킹(swapped packed) 네트워크를 통해 전송될 수 있다.

본 발명은 어플리케이션 설계자가 전화나 WWW 브라우저를 통하던지 액세스의 모드에 무관하게 하나의 프레임워크를 사용하여 어플리케이션을 셋업할 수 있게 한다. 어플리케이션 및 클라이언트 사이의 모든 대화는 html 및 적절한 것으로서 그 확장을 통해 표시되는 정보를 갖는 HTTP 프로토콜로 표준화되어 있다. WWW 서버 상의 어플리케이션은 어

플리케이션(전화, PC 브라우저 또는 다른 네트워킹된 고객 장치)에 액세스하는 클라이언트의 유형에 액세스하고, 클라이언트에 표시되는 정보는 적절하게 구성될 수 있다. 어플리케이션은 클라이언트 액세스를 위한 하나의 표준 프로토콜만을 지원할 필요가 있다. 또한, 어플리케이션 및 내용은 설계, 유지 및 변형이 용이한 일정한 프레임워크로 표시된다.

본 발명의 또 다른 양태에서, 대화식 브라우저는 XML 사양에 따른 대화식 마크업 언어(CML)를 해석한다. CML은 새롭게 경험되는 어플리케이션 개발자가 대화식 다이얼로그를 고속으로 개발할 수 있게 한다. 또 다른 양태에서, CML은 다른 서술 구분 또는 방법을 따를 수 있다. HTML 및 월드 와이드 웹과의 유사성을 더욱 따르면, CML 및 대화식 브라우저는 유산 기업 어플리케이션 및 유산 데이터베이스 주위에 대화식 사용자 인터페이스를 생성하기 위해 간단하고 체계적인 방법을 제공한다.

CML 파일/문서는 표준 네트워킹 프로토콜을 사용하여 HTTP 서버로부터 액세스될 수 있다. CML 페이지는 대화식 브라우저를 통해 사용자에게 제공될 대화식 UI를 나타낸다. 바람직하게는, CML 페이지는 XML 어플리케이션에 기초하는 태그에 의해 정의된다. 주요 구성 요소는 < page>, < body>, < menu> 및 < form> 이다. 페이지는 다른 CML 구성 요소를 그룹화하고, (XML에 의해 요구되는 바와 같이) CML 문서에 대한 최상위 레벨 구성 요소로서 작용한다. 본문(body)은 브라우저에 의해 구두로 전달될 출력을 지정한다. 메뉴는 사용자에게 선택의 목록을 제공하고, 각 선택을 사용자가 그러한 선택을 선택하는 경우 방문하도록 CML 구성 요소를 식별하는 URL과 결합시킨다. 형식은 사용자가 하나 이상의 정보를 제공할 수 있게 하고, 여기에서 각 정보의 내용은 문법에 의해 기술된다. 형식 구성 요소는 또한 사용자가 형식을 완성할 때 방문하도록 URL을 지정한다.

또 다른 양태에서, 대화식 마크업 언어 규칙이 내용 제공자에 의해 대화식 브라우저의 전체 장점을 취하도록 HTML 파일(또는 HTML 대신 사용되는 파일)에 추가될 수 있다.

또 다른 양태에서, 대화식 트랜스코더는 표시 재료를 하나의 양식으로부터 대화식 양식(일반적으로, 음성만 및/또는 음성 및 GUI)으로 변환한다. 이것은 GUI의 하나의 페이지를 CUI(대화식 사용자 인터페이스)의 하나의 페이지로 변환하기 위한 기능 변환뿐만 아니라 어플리케이션, 트랜잭션 또는 사이트의 비즈니스 논리를 허용 가능한 다이얼로그로 변환하는 논리 트랜스코딩을 수반한다. 대화식 트랜스코딩은 HTML 파일을 대화식 브라우저에 의해 해석되는 CML 파일로 변환할 수 있다. 트랜스코더는 서버, 브라우저 또는 내용 제공자의 특허 어플리케이션일 수 있다.

또 다른 양태에서, HTML/GUI 기반 구조 골격이 GUI 사이트의 다이얼로그 논리 또는 비즈니스 논리를 포착하는데 사용될 수 있다. 이 정보는 사이트, 논리 또는 어플리케이션을 안내하는 데 사용될 수 있다. 다이얼로그 흐름의 적절한 구성 후에, 각 구성 요소는 음성 전용 내용 또는 다중 양식(동기화 GUI 및 음성 인터페이스) 페이지로의 기능적인 트랜스코딩이 이루어질 수 있다.

또 다른 양태에서, 대화식 프록시가 예컨대, 장치, 브라우저 및/또는 엔진의 자격, 사용자 또는 어플리케이션의 선호도, 서버 상의 로드, 네트워크 상의 트래픽, 대화식 인수(데이터 파일)의 위치에 대하여 어플리케이션, 논리 또는 사이트의 내용 설명을 변경 및/또는 준비하기 위해 제공된다. 예컨대, 대화식 프록시는 법인 소프트웨어의 스크린 맵과 같은 독점 포맷을 직접 변환시킬 수 있다.

본 발명의 이들 및 다른 양태, 특징 및 장점은 첨부하는 도면과 관련하여 이하의 바람직한 실시예의 상세한 설명으로부터 보다 명확히 이해할 수 있을 것이다.

#### 도면의 간단한 설명



도 1은 본 발명의 바람직한 실시예에 따른 대화식 브라우징 시스템을 도시한 도면.

도 2는 본 발명의 일 실시예에 따른 대화식 브라우징 시스템을 실행하는 정보를 액세스하는 시스템의 블록도.

도 3은 본 발명의 다른 실시예에 따른 대화식 브라우징 시스템을 실행하는 정보를 액세스하는 시스템의 블록도.

도 4a는 본 발명의 일 실시예에 따른 대화식 브라우징 시스템을 실행하는 정보를 액세스하는 분배 시스템을 도시한 블록도.

도 4b는 본 발명의 다른 실시예에 따른 대화식 브라우징 시스템을 실행하는 정보를 액세스하는 분배 시스템을 도시한 블록도.

도 5는 본 발명의 일 실시예에 따른 대화식 마크업 언어를 사용하는 대화식 정보 액세스 시스템의 블록도.

도 6은 본 발명의 일 실시예에 따른 대화식 마크업 언어를 사용하는 분배 대화식 시스템의 블록도.

도 7은 본 발명의 일 실시예에 따른 대화식 마크업 언어를 사용하는 예시적인 분배 대화식 시스템의 블록도.

도 8은 본 발명의 다른 실시예에 따른 대화식 마크업 언어를 사용하는 다른 분배 대화식 시스템을 도시한 도면.

도 9는 본 발명의 일 실시예에 따른 대화식 마크업 언어를 사용하는 또 다른 분배 대화식 정보 액세스 시스템을 도시한 도면.

도 10은 본 발명의 일 실시예에 따른 대화식 마크업 언어를 사용하는 또 다른 예시적인 분배 대화식 정보 액세스 시스템을 도시한 도면.

#### 실시예

본 발명은 대화식 브라우징 시스템 및 상호 연결된 CML 페이지의 세트를 사용하여 대화식 브라우징 시스템을 구성하기 위한 CML(대화식 마크업 언어)에 관한 것이다. 이러한 대화식 시스템은 HTML(하이퍼텍스트 마크업 언어)을 사용하여 기록된 한 세트의 상호 연결된 페이지를 사용하여 대화식 GUI 브라우저 어플리케이션을 구성하는 것과 개념적으로 유사하다. 더욱이, HTML이 GUI 동작을 다른 페이지를 방문하거나 서버와 통신하는 것과 같은 어플리케이션 동작으로 해석하기 위한 한 세트의 메카니즘을 제공할 때, 대화식 브라우저 및 CML이 구두 입력을 유사한 어플리케이션 동작으로 해석하기 위해 사용된다. CML 페이지는 대화식 브라우저를 통해 사용자에게 해석 및 표시될 대화식 UI를 나타낸다. 바람직하게, CML 페이지는 (이하 더욱 상세히 설명되는 바와 같이) 현재의 XML(확장 가능 마크업 언어) 어플리케이션에 기초하는 태그로 정의된다.

본 명세서에 설명되어 있는 대화식 시스템 및 방법은 다양한 형태의 하드웨어, 소프트웨어, 펌웨어, 특수 용도 프로세서 또는 이들의 조합 형태로 수행될 수 있다. 특히, 대화식 브라우저는 바람직하게 프로그램 기억 장치(예컨대, 자기 플로피 디스크, RAM, ROM, CD ROM 및/또는 플래시 메모리)에서 명백히 구현되고, 퍼스널 컴퓨터와 같은 적절한 구조를 포함하는 임의의 장치 또는 머신과 PDA 및 스마트 폰과 같은 보급형 계산 장치에 의해 실행 가능한 프로그램 명령을 포함하는 어플리케이션으로서 수행된다.

대화식 브라우저의 구성 요소중 일부 및 첨부하는 도면에 도시되어 있는 다른 시스템 구성 요소는 바람직하게 소프트웨어로 수행되기 때문에, 그러한 구성 요소 사이의 실제 접속은 본 발명이 프로그램되는 방식에 따라 상이할 수 있다. 본 명세서에 개시된 기술적 사상에 의해, 당업자라면 본 발명의 이들 및 유사한 수행 또는 구성을 예측할 수 있을 것이다.

## 대화식 브라우저 구조

이하, 도 1을 참조하면, 블록도는 본 발명의 바람직한 실시예에 따른 대화식 브라우저 시스템을 도시한다. 통상적으로, 대화식 브라우저 시스템(10)은 사용자가 HTTP와 같은 표준 네트워킹 프로토콜을 사용하여 음성 입력(유일하게 또는 DTMF, 키보드, 마우스 등과 같은 다른 양식과 관련하여)을 통해 유산 정보 서비스 및 트랜잭션에 액세스할 수 있게 한다. 또한, HTTP 프로토콜은 본 발명의 하나의 바람직한 실시예이지만, 다른 유사한 프로토콜이 유리하게 사용될 수도 있다. 예를 들어, 이것은 TCP/IP, WAP(무선 어플리케이션 프로토콜), GSM, VoIP 등과 같은 임의의 프로토콜 또는 IP를 지원해서 TCP/IP 또는 유사한 특징을 제공하는 임의의 다른 프로토콜의 상부에 배치될 수 있다. 더욱 일반적으로, TCP/IP가 사용 불가능한 경우, TCP/IP와 유사한 특징을 제공하는 다른 프로토콜을 수행하거나, 적어도 네트워크 및 전송 기능(본 발명은 전송 및 네트워크 계층에 무관하다)을 실행할 수 있다.

도 1에서, 본 발명의 일 실시예에 따른 대화식 브라우저 시스템(10)은 CVM(대화식 가상 머신) 시스템(15)의 상부에서 실행하는 대화식 브라우저(11)(대화식 어플리케이션)를 포함한다. 대화식 브라우저(11)는 일반적으로 HTML 또는 DB2와 같은 종래의(유산) 구조의 문서 포맷을 중간 문서 또는 (후술하는 바와 같은) 미리 지정된 트랜스코딩 규칙에 따른 CML(대화식 마크업 언어) 문서로 트랜스코딩하는 트랜스코더 모듈(11)을 포함한다. CML 문서는 사용자에게 제공될 유산 정보 포맷의 대화식 UI를 나타낸다. 특히, CML 문서는 예컨대, HTML 기반 정보를 합성 오디오 메시지로서 사용자에게 제공하도록 CML 분석기/프로세서(14)에 의해 처리되는 메타(meta) 정보를 포함한다. CML 문서의 다양한 실시예가 본 발명에 의한 수행에서 예측된다. 이하 상세히 설명되는 바람직한 실시예에서, CML 문서는 XML(확장 가능 마크업 언어)에 기초하는 태그에 의해 정의된다. 그러나, CML을 수행하는 임의의 명백한 방법이 이용될 수도 있다. XML이 간단함, 파워 및 현재의 대중성으로 인해 현재 바람직하다.

대화식 브라우저(11)는 음성 명령, DTMF 신호 및 키보드 입력과 같은 사용자 명령(다중 양식) 입력을 HTTP 요구와 같은 네비게이션 요구(navigation request)로 변환시키는 명령/요구 프로세서(12)(명령 및 제어 인터페이스)를 더 포함한다. 순 음성 대화식 브라우저에서는, 입력은 음성만이다. 그러나, 대화식 브라우저(11)는 다중 양식 입력용으로 구성될 수 있다.

특정 대화식 기능 또는 서비스가 요구될 때, 대화식 브라우저(11)는 (이하에 설명하는 바와 같이) 그러한 서비스를 요구하는 CVM(15)에 API 호출을 생성한다. 예를 들어, CML 문서를 (CML 분석기/프로세서(14)를 통해) 해석할 때, 대화식 브라우저(11)는 사용자에게 합성 음성 출력을 제공하기 위해 (CVM(15)을 통해) TTS(텍스트-음성 합성) 엔진(26)에 훅(hook)할 수 있다. 또한, 음성 명령 또는 자연 언어 질문(예컨대, 네비게이션 요구)이 입력될 때, 대화식 브라우저(11)는 그러한 입력 명령을 처리하기 위해 음성 인식 엔진(24) 및 NLU(자연 언어 이해) 엔진(25)에 훅할 수 있으며, 그것에 의해 명령/요구 프로세서가 예컨대, 적절한 HTTP 요구를 발생시킬 수 있게 한다. CVM 시스템(15)은 임의의 종래의 OS(운영 체제) 또는 RTOS(실시간 운영 체제) 상에서 실행할 수 있는 쉘(shell)이다. CVM 시스템(15)의 구조 및 동작의 상세한 논의는 본원과 동시에 출원되어 공동 양수되어 참고로 본 명세서에 완전히 통합되어 있는 "Conversational Computing Via Conversational Virtual Machine" 라는 명칭의 특허 출원 IBM Docket No. YO99 9-111P에 제공되어 있다.

간략하게, 도 1에 도시되어 있는 바와 같이, CVM 시스템(15)은 대화식 API 계층(16)을 포함하며, 이를 통해 대화식 브라우저(11)가 대화식 엔진(23)을 포함하는 특정 대화식 서비스 및 동작에 (시스템 호출을 통해) 액세스하도록 CVM 커널 계층(17)에 통화할 수 있다. CVM 커널(17)은 엔진 및 인수(국부 및/또는 분배 중 하나)와 같은 대화식 자원을 할당하고, 범용 및 조정 CUI(대화식 사용자 인터페이스)를 제공하도록 등록된 대화 자격 및 요건에 기초하여 어플리케이션 및 장치(국부 및/또는 분배) 간의 다이얼로그 및 문맥을 관리 및 제어할 책임이 있다. CVM 시스템(15)은 대화식 서비스를 실행하고, 국부 대화식 엔진(23), 예컨대 음성 인식 엔진(24), NLU(자연 언어 이해) 엔진(25), TTS

(텍스트 - 음성) 엔진(26) (뿐만아니라 NLG(자연 언어 발생) 엔진, 스피커 인식 엔진과 같은 다른 엔진)에 (SAPI, S RAPI, JSAPI, SVAPI 또는 그러한 엔진 API의 확장과 같은) 대화식 엔진 API 계층(18)을 통해 호출을 수행함으로써 기능을 한다. 또한, 엔진 호출은 분배 토폴로지에서의 원격 음성 엔진으로 행해질 수 있다. 또한, (오디오 포착, 압축, 압축 해제 및 재구성을 제공하는) 오디오 서브시스템(33) 및 (키보드(28), 마우스(29), 터치 스크린, 마이크로폰/스피커(31) 및 DTMF(32)와 같은) 임의의 다른 다중 양식 I/O 자원으로의 호출이 종래의 드라이버/API 계층(22)을 통해 실행된다.

CVM 시스템(15)은 TCP/IP, HTTP, WAP 등과 같은 종래의 프로토콜(20)을 통해 네트워크 통신을 제공하는 통신 스택(19)을 더 포함한다. 통신 스택(19)은 분배 어플리케이션에 이용되는 대화식 프로토콜(21) (또는 분배 대화식 프로토콜)을 더 포함한다. 상기 통합되어 있는 IBM Docket No. YO999 - 111P에 기재되어 있는 바와 같이, 대화식 프로토콜(또는 방법) (21)은 (1) 대화로 인식되는(즉, 대화식 프로토콜로 통신하는) 네트워크 장치 및 어플리케이션을 찾기 위한 프로토콜; (2) 네트워크 장치 사이의 대화식 엔진 및 인수와 같은 대화 자격(자원)을 등록하기 위한 프로토콜; (3) 등록된 대화 자격에 기초하여 (마스터/슬레이브, 통신 실체간(peer - to - peer)과 같은) 네트워크 구조를 네비게이션하기 위한 프로토콜; (4) 다이얼로그, 대화식 인수, 애플릿, 액티브X 구성 요소, 절차적인 객체 및 다른 실행 가능 코드의 상태, 문맥 및 이력에 관한 정보와 같은 네트워크 접속 장치 사이의 대화를 조정하도록 정보를 교환하기 위한 프로토콜; (5) 압축 음성(파형 또는 특징)을 송신 및 수신하도록 음성 코딩 프로토콜을 위한 프로토콜을 포함한다. 이들 대화식 프로토콜(21) 뿐만 아니라 네트워킹된 장치 사이에 대화 조정을 제공하기 위한 규칙은 본원과 동시에 출원되어 공동 양수되어 참고로 본 명세서에 통합되어 있는 " System and Method For Providing Network Coordinated Conversational Services" 라는 명칭의 특허 출원 IBM Docket No. YO999 - 113P에 또한 개시되어 있다.

도 1에 도시되어 있는 엔진(23) 및 I/O 자원(27), 대화식 드라이버/API(22) 및 오디오 서브시스템(33)은 대화식 브라우저(11) 및 CVM 시스템(15)이 실행되는 하위 장치, 머신 또는 플랫폼의 일부인 구성 요소이다. 더욱이, 대화식 브라우저(11) 및 CVM 시스템(15)은 별개의 시스템으로서 제공될 수 있거나, 다른 방법으로서, 대화식 브라우저(11)는 자체의 CVM 시스템(15)을 포함하는 자립형 어플리케이션으로서 수행될 수 있다(이 경우에 브라우저 및 CVM 플랫폼은 동일, 즉 구별할 수 없는 엔티티이다). 또한, 특별히 전술한 바와 같이, CVM 시스템(15)이 없을 때에는, 대화식 브라우저(11)가 CVM 시스템(15)에 대하여 상기 전술한 것과 유사한 기능을 수행한다(예컨대, 대화식 브라우저가 API 호출을 적절한 엔진에 국부적으로 및/또는 분배하여 생성한다). 실제로, API, 서비스, 특징, 동작, 엔진으로의 액세스 및 통신 메카니즘은 모두 브라우저에 의해 제공되는 특징 및 서비스의 일부로서 대화식 브라우저 어플리케이션(11)에 직접 생성되거나 그 일부에 생성될 수 있다(이것은 넷스케이프 브라우저에 하위 운영 체제로 제공되는 JVM(자바 가상 머신)을 갖거나 자체의 JVM을 수행하는 브라우저를 갖는 것과 유사하다). 또한, 대화식 브라우저는 브라우저에 병렬로 실행하는 하위 운영 체제 및/또는 어플리케이션에 의해 제공되는 유사한 기존의 서비스의 장점을 취할 수 있다.

트랜스코더(13)는 도 1에 도시되어 있는 바와 같이 대화식 브라우저(11)의 일부일 수 있다. 다른 방법으로, 이하에 설명하는 바와 같이, 트랜스코더(13)는 네트워크에 위치될 수 있고, 예컨대, 내용 제공자의 서버(서버측) 상의 대화식 프록시 서버 또는 네트워크내의 여러 머신 상에 분배된 다른 머신에서 실행된다. 또한, 트랜스코더는 프록시 서버, 브라우저 또는 내용 제공자의 특허 어플리케이션일 수 있다. 트랜스코더는 또한 통합 소프트웨어의 스크린 맵과 같은 다른 특허 포맷을 직접 변환할 수 있다. 트랜스코더(13)는 예컨대, HTML 문서(또는 다른 포맷 페이지)를 CML 문서로 변환하는 기능 트랜스코딩 모듈(13a)를 포함한다. 이러한 변환을 수행하는 다양한 규칙을 이하에서 설명한다. 또한, 트랜스코더(13)는 의미있는 다이얼로그를 생성하도록 어플리케이션 뒤의 논리 정보를 액세스 및 해석하는 논리 트랜스코딩 모듈(13b)을 포함한다. 이것은 종종 어플리케이션 및/또는 비즈니스 논리에 관한 정보를 광범위하게 요구한다. 논리 트랜스코딩을 실행하는데 이용될 수 있는 다른 방법이 본원과 동시에 출원되어 공동 양수되어 참고로 본 명세서에 통합되어 있는 " Structure Skeletons For Efficient Voice Navigation Through Generic Hierarchical Objects" 라는 명칭의 특허 출원 IBM Docket No. YO99 - 114에 개시되어 있다. 이 출원은 대화식 구조 골격이 논리 트랜스코딩 정보를 인코딩 및 송신하는데 사용될 수 있는 방법을 기재하고 있다.

바람직하게, 트랜스코더(35)는 임의의 트랜잭션 형식 또는 HTML 페이지를 CML 페이지로 변환할 수 있는 일반적인 범용 변환기이다. 형식이 채워진(HTML 또는 CGI) HTML 다이얼로그 페이지 또는 ATM 다이얼로그(및 다른 3270 유형의 스크린 다이얼로그)가 제한된 정보를 갖거나 제한된 사용자 선택을 제공하는 ATM 다이얼로그 또는 IVR 다이얼로그와 같은 직통식 다이얼로그에 대해서, 트랜스코딩은 CML 페이지에 의해 쉽게 공식화될 수 있다. 다른 복잡한 HTML 문서는 대신에 변환을 위한 트랜스코더를 필요로 함이 없이 (이하 설명되는 바와 같이) 첫번째 서버측에 CML로 서 태그될 수 있다. 실제로, 이하에 설명하는 바와 같이, 바람직한 실시예에서, 대화식 브라우저에 제공되는 모든 내용은 예컨대, 브라우저 또는 서버측 트랜스코더에 의해 플라이(fly) 상에서 트랜스코딩되는 일반적인 HTML 페이지와는 대조적으로, 트랜스코더 또는 트랜스코딩 프록시가 필요없이 CML에 직접 기록된다. 이와 달리, 기본 구성 및 설계 규칙이 HTML에서 CML로의 변환을 간략하게 하기 위해 부과된다. 예를 들어, 뉴스 및 e-메일/음성 메일은 인토네이션(intonation) 및 JSAPI식 TTS 마크업 정보를 태그로 부착하기 전에, 종래의 TTS 필터에 의해 입력 텍스트를 필터링함으로써 변환될 수 있다. 실제로, 이들 변환 규칙은 서비스 제공자 또는 표준의 대상에 기초하여 전용물로 될 수 있다. 이들 규칙은 다음을 포함할 수 있다:

1. HTML 페이지가 이미지를 포함하는 경우, 트랜스코더는 이미지를 폐기하고, 그러한 정보를 가청적으로 표시하는 이미지 태그에 (만약 있다면) 부가되는 이미지명 및 명령을 나타내는 필요한 메타 정보를 발생한다.

2. HTML 페이지가 가청적으로 표시될 수 있는 구성 요소가 없는 스크립트 및/또는 애플릿을 포함하는 경우, 트랜스코더는 특정 경고 메시지를 실행하는 메타 정보를 발생할 수 있다. 경고는 여러 용도를 갖는다. 예를 들어, 경고 메시지는 사용자에게 HTML 페이지가 스크립트 또는 애플릿을 포함하는 것을 통지할 수 있다. 실제로, 스크립트 또는 애플릿에 의해 제공되는 임의의 음성 출력이 없을 때, 사용자는 (애플릿 애니메이션이 즉시 시청 가능한 GUI 브라우저와 대조적으로) 스크립트 또는 애플릿에 관하여 알 수 없다. 또한, 경고 메시지는 사용자에게 의미있는 CUI(대화식 사용자 인터페이스) 특징(이것은 애플릿이 CUI 구성 요소를 통합하도록 CVM 라이브러리(상기 통합되어 있는 IBM Docket No. YO999-111P에 기재되어 있는 것과 같은)의 일부분으로서 대화식 기본 등급을 가지고 기록되는 대신에, 자바와 같은 종래의 GUI 언어로 기록되는 상황에 대해서도 참이다)으로 트랜스코딩될 수 없는 애플릿에 포함되는 어떤 정보/내용을 상실할 수 있음을 통지하는 작용을 한다. 또한, 메타 정보는 1999년 4월 7일에 출원되어 공동 양수되어 참고로 본 명세서에 통합되어 있는 "Multi-Modal Shell" 이란 명칭의 미국 가출원 60/128,081호에 기재되어 있는 것과 같이, 다중 양식 브라우저의 GUI(시청)측과 음성측을 동기화시키는데 사용될 수 있다. 상기 미국 가출원의 발명에 따르면, 절차적인 구성 요소(애플릿 및 액티브X 구성 요소와 같은)가 장치/브라우저의 자격에 기초하여 GUI만 또는 음성만(및 이것의 변형)으로 트랜스코딩될 수 있다. 이것은 어플리케이션 개발자 또는 구성 요소(기본 등급)의 창작자에 의해 자신의 절차에 사용되는 어플리케이션을 고유적으로 코딩된 트랜스코딩의 문제점을 자동으로 해결한다.

3. 프레임을 포함하는 HTML 페이지를 이용하면, 트랜스코더는 먼저 자신의 명칭 또는 인덱스 번호로 프레임을 나타내는 메타 정보를 발생할 수 있다. 예를 들어, 오디오 재생은 "이 페이지에는 4 개의 프레임이 있다: 프레임 1은 ..을 나타내고..." 일 수 있다. HTML 및 CML에 의해 생성되는 모든 프레임 링크 및 기능은 전체 페이지가 다운로드되자마자 활성화될 수 있다. 이것은 프레임이 메뉴로서 사용될 수 있게 한다. 다른 방법으로는, 메인 프레임만 표시되도록 규칙이 적용될 수 있지만, 나머지 프레임은 판독하진 않지만 활성화된다.

4. HTML 페이지는 JSAPI 마크업(또는 유사한 변형)을 생성하도록 TTS 필터를 사용하여 필터링될 수 있다.

5. HTML 페이지내의 임의의 하이퍼링크는 기본 형식 및 FSG(유한 상태 문법)으로 변환될 수 있다. 특히, 하이퍼링크 및 다른 명령 및 사용자가(음성 명령에 의해) 따를 수 있는 입력이 브라우저의 명령의 활성 목록에 부가될 수 있다. 이것은 어휘, 필요할 때 이들 워드의 발음 규칙(즉, 기본 양식)을 부가하고, 이들 새로운 명령을 지원하기 위해 적절하게 문법 또는 NLU 프레임워크/문맥을 변경하는 것을 수반한다.

6. 텍스트는 상이한 폰트에 대하여 상이한 음성으로 합성될 수 있다. 그러나, 대화식 브라우저는 음성을 변화시키도록 선택할 수 없다.
7. 하이퍼링크는 오디오 아이콘(사운드 또는 다른 음성 또는 배경 음악)으로 표시될 수 없다.
8. 각 하이퍼링크는 예컨대, < !··CML 태그··> 로 표시되는 경우, 다음 페이지로 이동할 수 있는 숨겨진 메뉴로서 표시될 수도 있다.
9. 풀 다운 메뉴 및 제한된 메뉴가 적절한 어휘 및 명령을 갖는 다이얼로그로 자동으로 변환될 수 있다.

10. 프리 필드는 NLU 또는 구술 능력을 활성화시킬 수 있다. 통상적으로, 범용 구술에 있어서, 사용되는 음성 인식 엔진은 대화식 브라우저의 국부 디폴트 엔진일 수 있다. 다른 방법으로는, 분배 시스템에 있어서(후술하는 바와 같이), 내용 제공자는 브라우저가 필요한 자격을 갖지 않거나 원격 음성 인식 엔진이 NLU 또는 구술 어플리케이션에 특별히 동조되는 경우, 음성 인식 엔진의 제안된 URL을 제공할 수 있다. NLU 형식으로 채워진 NLU 다이얼로그는 HTML 페이지에서의 CML 명령 태그로서 NLU 엔진의 적절한 URL을 통과시키거나 HTML 페이지 대신에 CML 페이지를 직접 제공함으로써(이하 더욱 상세히 설명되는 바와 같음) 실행될 수 있다.

11. 다중 형식 기반 스크린 기반 트랜잭션 GUI 다이얼로그는 각 메뉴가 자립형 구성 요소(예컨대, 어드레스를 얻거나 아이템을 주문하는 것과 같은 완전한 작업을 실행하는)인 경우 연속적인 다이얼로그의 메뉴로 직접 변환될 수 있다.

대화식 브라우저(11)는 또한 사용자가 주의 위드를 표현할 수 있게 하거나, (이하 더욱 상세히 설명되는 바와 같이) 새로운 명령 또는 입력을 제공하기 전에 TTS 표시의 출력 흐름을 차단할 수 있게 하는 "가담(barge-in)" 특징을 포함할 수 있다. 이것은 사람이 말을 할 때 누르는 버튼에 의해 수행될 수 있으며, 그것에 의해 버튼이 통화하도록 눌러지거나 시작하도록 눌러지고/눌러지거나 정지하도록 눌러진다.

바람직하게는, 트랜스코더에 의해 발생하는 CML 문서(또는 후술하는 바와 같이 내용 제공자에 의해 직접 제공되는 CML 문서)는 아래에 예시한 특징 및 이러한 특징을 나타내는데 필요한 태그의 임의의 조합을 포함한다:

1. 사용자에게 말해지거나 그로부터 숨겨진 활성 링크;
2. 사용자에 의해 페이지에 제공되는 가능한 입력을 디코드하는데 사용될 수 있는 대화식 인수(예컨대, 어휘, 문법 및 기본 형식의 데이터 파일)로의 링크;
3. 분배 시스템에 있어서, 사용자 입력을 디코드하고/하거나 오디오 출력을 발생하기 위해 소정의 페이지를 전송하는 위치를 제공하는데 사용될 수 있는 음성 인식 URL, 스피커 인식 URL, TTS URL 및/또는 NLU 엔진 URL 등으로의 링크. 국부 호스트가 국부 (디폴트) 엔진을 부과하는데 사용될 수 있다. 따라서, URL이 제공되지 않는 경우, 브라우저 디폴트 엔진이 사용된다. 이것은 브라우저, 장치, 사용자 선호도 또는 서비스 제공자(ISP 또는 ASP) 선호도에 기초하여 설정될 수 있다. 이것은 브라우저 디폴트 위치를 설정하기 위해 브라우저상 또는 서버상의 임의의 관련 메타 정보를 사용하는 것과 같다;
4. 사용자에게 재생하기 위한 오디오 파일에 대한 URL(예컨대, IVR 프롬프트);
5. 소정의 링크에 따르기 전에 어떤 입력에 대한 사용자 확인을 부과하는 확인 요구 태그;
6. 소정의 링크에 관련된 DTMF 코드, 뿐만 아니라 복잡한 키 엔트리에 대한 DTMF 문법;

7. 텍스트가 사용자에게 어떻게 재생되는지를 나타내는 TTS 마크업(JSAPI 식);

8. 자신의 다이얼로그를 지향하는 대화식 어플리케이션을 실행하기 위한 스크립팅 언어 및 절차 코드(자바, 자바스크립트, 액티브X 등과 같음). (후술하는 바와 같이) 브라우저가 CVM의 상부에 형성될 때 대화식 프로그래밍 언어 및 기본 등급(즉, CVM 라이브러리)가 사용될 수 있다. 이와 달리, 코드 및 서비스가 브라우저, 하위 운영 체제 또는 브라우저와 병렬로 사용할 수 있는 다른 어플리케이션 또는 서비스에 의해 제공될 수 있다.

9. 자원에 대한 브라우저 최소 요건(예컨대, 스크립트, 이미지 등)을 나타내는 대화 자격 태그. 예를 들어, 비오디오 인 에이블 스크립트는 다운로드되지 않거나, 그러한 스크립트가 의미있는 작업을 실행하는 경우를 제외하고는 음성 전용 장치에서 실행된다. 또한, (이미지가 다중 양식 및 GUI 전용 브라우저에 관련되더라도) 이미지는 음성 전용 브라우저 상에 다운로드되지 않는다. 표시 자료가 브라우저(즉, 하위 엔진)의 능력에 따라서 변환될 때, 변환을 실행하는 엔티티는 대화식 프록시이다;

10. 어떤 특정 처리(예컨대, 스피커 인식 단계 또는 압축 루틴에 있어서의 암호 플러그 인 및 특수 엔진에 있어서의 전단부)를 위해 사용/다운로드하기 위한 플러그 인의 URL; 및/또는

11. 호출될 때 가능한 링크의 목록 상의 페이지를 따르도록 제안하는 페이지(예컨대, 자체에 대응하는 기본 형식, 문법 등을 갖는 키워드의 세트)를 북마킹 및 이후에 그 페이지로 복귀하기 위한 메타 정보 태그. 이것은 일반적으로 (상기 통합되어 있는 IBM Docket No. YO999-111P에서 논의된 바와 같이) CVM에 의해 제공되는 것과 같은 분류/메타 정보 서비스에 관련된 특징이다.

특정 명령 < !···CML 프록시 url=...··· > 태그가 원하는 트랜스코딩 기능(대화식 프록시, 기능 트랜스코딩 또는 논리 트랜스코딩)을 실행하기 위해 주문 대화식 프록시 또는 트랜스코딩을 지정하도록 HTML 페이지에 통합될 수 있다. 다른 방법으로는, (브라우저, 서버 또는 분배 프록시의 일부인) 디폴트 트랜스코더에 의해 변환이 실행될 수 있다. CML 페이지는 또한 대화식 브라우저의 능력에 기초하여 (프록시) 트랜스코더에 의해 주문될 수 있다. 이러한 기능은 URL 태그를 프록시로서 사용하는 것과 상이하게 실행될 수 있다(이것은 브라우저가 프록시 서버에 접속하여 프록시를 통해 HTTP 요구를 전송하는 것을 의미한다). CVM이 제공될 때, 하위 핸드셰이크가 브라우저용의 CVM에 사용 가능한 엔진의 대화 능력에 관한 메타 정보를 교환하기 위해 발생할 수 있고, 프록시가 페이지에 자동으로 주문할 수 있는 것에 기초하여 발생할 수 있다. 또한, 대안은 그러한 경우에 엔진 능력에 적합한 상이한 페이지로 전송되는 페이지를 갖고 이루어진다.

또한, < !···CML 태그...··· > 는 대화식 브라우저 또는 대화식 프록시에 의해 직접 액세스되는 HTML 또는 XML 페이지에 직접 CML 태그를 도입할 수 있다. 이들 CML 특징은 각각 이하 더욱 상세히 설명된다.

CVM은 특정 브라우저/클라이언트 자격(예컨대, IVR)에 대하여 다중 양식이기 때문에, 관련 기능만이 태그된다. 바람직한 실시예에서, 브라우저는 일반적으로 ML의 일관성에 영향을 주는 대신에 비공지 태그를 무시한다. 더욱이, 범용 시스템에 있어서, 브라우저는 쿠키(cookie) 메카니즘을 사용하거나 내용 제공자의 첫번째 페이지에 또는 서버 게이트웨이/포털에 접속할 때 숨겨진 형식을 채워서 프록시 또는 서버에 자신의 자격을 등록하여야 한다. 쿠키는 사용자를 식별 또는 인증하는데 필요한 정보를 포함할 수 있다. 쿠키는 또한 브라우저의 현재의 다이얼로그 상태에 관한 모든 문맥 정보를 포함할 수 있다. 이것은 또한 브라우저에 접속 시에 서버와 공유되거나 전송되는 국부 문맥, 이력 및/또는 선호도 정보의 나머지를 포함할 수 있다. 쿠키 또는 다른 등록 프로토콜을 통해 다이얼로그 정보, 문맥 및 다른 메타 정보를 서버에 전송하는 자격은 서버가 즉시 대화에 참여할 수 있게 한다. 다른 등록 메카니즘이 쿠키 메카니즘과 다르게 수행될 수 있다.

대화식 마크업 언어 규칙이 내용 제공자에 의해 HTML 파일에 부가되거나 대화식 브라우저의 모든 장점을 취하기 위해 HTML 파일을 대체할 수 있다. 이러한 언어는 (이하 설명되는 바와 같이) 이전의 섹션에 설명되어 있는 특징을 나타내도록 태그를 제공한다.

#### 대화식 브라우저를 사용하는 전화 어플리케이션

이하, 도 2를 참조하면, 블록도는 본 발명의 일 실시예에 따른 전화 서비스를 제공하는 대화식 시스템을 도시한다. 이 시스템(100)은 클라이언트 장치(102)에 의해 전화 네트워크(104)(예컨대, PSTN(공중 교환 전화망)) 상으로 액세스 가능한 IVR(101)(또는 호출 센터)를 포함한다. IVR(101)은 네트워크(104)상의 하나 이상의 내용 서버(105)에 접속된다. 이 실시예에서, IVR(101)은 전화 회사(텔코)의 구내의 머신에서 실행하는 것으로 가정한다. IVR(101)은 예컨대, ATM 전화 서비스를 제공하도록 금융 어플리케이션의 서비스 제공자와 결합될 수 있다. 클라이언트 장치(102)는 예컨대, 종래의 전화, 이동 전화 또는 전화 네트워크(103)를 통해 IVR(101)에 통화를 설정하기 위해 전화 카드를 갖는 PC일 수 있다. 네트워크(104)는 인터넷, 인트라넷, LAN, 특수 목적 네트워크 등일 수 있다. 내용 서버(105)는 예컨대, IVR(101) ATM 서비스와 결합되는 내용 페이지(HTML, CML 등)를 포함하는 HTTP 서버일 수 있다.

IVR(101)은 CVM(106a)을 실행하는 자립형 어플리케이션으로서 도시되어 있는 대화식 브라우저(106)를 포함한다. 도 2의 대화식 브라우저(106) 및 CVM(106a)은 도 1에 대하여 기술한 대화식 브라우저(11) 및 CVM(15)과 구조적으로 및 기능이 유사하다(그것에 의해 CVM은 브라우저 내에 또는 하위 운영 체제에 포함될 수 있다). IVR 시스템(100)은 오디오 포착, 음향 전단부(음향 처리) 및 오디오 서브시스템(예컨대, 사운드카드)와 같은 오디오 기능을 제공하는 오디오 시스템(107)을 더 포함한다. IVR(101)은 DTMF 프로세서(108) 및 대화식 엔진(109)(예컨대, 음성 NLU, TTS 엔진 등)을 더 포함한다.

클라이언트(102)가 IVR(101)에 접속될 때, 음성 명령 및 DTMF 신호와 같은 입력 명령이 오디오 시스템(107)의 음향 전단부에 의해 수신되어 각각 적절한 대화식 엔진(109)(예컨대, 음성 엔진) 및 DTMF 프로세서(108)로 전송된다. 처리된 음성/DTMF 명령은 대화식 브라우저(106)에서 적절한 기능으로 호출로서 전달된다. 대화식 브라우저(106)는 차례로 네트워크(104)를 통해 하나 이상의 내용 서버(105)로부터 원하는 정보를 얻기 위해 적절한 요구(예컨대, HTTP 요구)를 발생한다. 내용 서버(105)는 적절한 정보를 검색하고(또는 지정된 동작을 완료하고) 이어서 결과, 메시지 및/또는 메뉴 옵션을 (전송한 TCP/IP 또는 유사한 프로토콜을 사용하여) HTML(또는 다른 유사한 포맷), CML, 압축 오디오 또는 이들의 조합으로 송신한다(정보가 내용 서버(105)에 구성되는 방법에 따름). 전송한 바와 같이, HTML(및 다른 구조 문서)에 있어서, 대화식 브라우저는 (전송한 CML 분석기/프로세서(14)를 통해) 처리하기 위한 CML 문서를 발생하도록 (전송한 트랜스코더(13)를 통해) 정보를 트랜스코딩하는 반면에, 서버(105)로부터의 CML 문서는 (트랜스코딩 없이) 직접 처리될 수 있다. 대화식 브라우저(106)는 사용자와의 다이얼로그를 설정하는데 필요한 대화식 엔진에 (CVM(106a)을 통해) 호환될 수 있다. 이하에 설명하는 바와 같이, 대화식 엔진은 네트워크(104)를 통해 분배될 수 있다.

대화식 브라우저는 또한 표준 네트워크 프로토콜(HTTP와 같은)을 사용하여 내용 서버(105)로부터 송신된 압축 오디오를 디코딩할 수 있고, 이어서 사용자에게 전화 오디오로서 오디오를 재생할 수 있다(그것에 의해 예컨대, 대화식 브라우저(106)를 통해 IP 전화가 가능하게 한다). 역으로, 오디오 입력은 오디오 시스템(107)의 오디오 포착부를 통해 포착될 수 있고, (상기 통합되어 있는 특허 출원 IBM Docket No. YO999-113P에 기재되어 있는 바와 같이) 분배 인식 및 처리를 위해 네트워크(104)를 통해 서버에 압축 또는 비압축 포맷으로 (대화식 브라우저(106)를 통해) 송신될 수 있다. 전송한 바와 같이, 압축 음성의 송신은 (YO999-111P 뿐만 아니라 YO999-113P에 기재되어 있는 바와 같이) CVM(106a)의 통신 스택에 의해 실행된다. CVM이 없을 때, 다른 통신 엔진이 핸드셰이크, 통신 및 코딩/디코딩을 실행한다. 그러한 통신은 HTTP 또는 VoIP에 제한되지 않으며, 임의의 TCP/IP 및 관련 프로토콜을 포함할 수 있다.

예를 들어, WAP, 블루투스(Bluetooth), 호핑 통신 프로토콜이 본 발명에 이용될 수 있다. 오디오가 국부적으로 포착되고 원격 서버로 이동될 때, 이것은 플러그 인(원시 수행)을 사용하여 또는 예컨대, 사용자의 입력을 포착하도록 오디오 및 멀티미디어 API를 사용하는 자바 애플릿 또는 자바 프로그램에 의해 실행될 수 있다. 또한, 대화식 통신(코딩) 프로토콜은 예컨대, (상기 통합되어 있는 YO999-111P 및 YO999-113P에 기재되어 있는 바와 같이) 소켓 접속, RCP, RMI 등을 사용하여 압축 음성을 송신하는데 사용될 수 있다.

도 2에 나타내는 시스템은 IVR 서비스를 제공하는 단 하나의 가능한 실시예이다. 또 다른 실시예에서, 예컨대, IVR은 사용자로부터의 입력 음성을 포착하도록 오디오 포착 기능을 제공하고 이어서 포착된 음성을 대화식 브라우저를 갖는 웹 서버로 송신하는 의미로 시스템의 "사운드카드"로서 단독으로 동작할 수 있다(유사하게, 서버로부터의 오디오 출력을 수신하여 사용자에게 재생하기 위해 IVR에 송신한다). 이 경우에, IVR은 사운드카드 또는 대화식 브라우저용 오디오 서브시스템으로서 동작한다. 다른 방법으로는, IVR에는 대화식 엔진(음성 인식 엔진, TTS 엔진, 스피커 인식 엔진 등) 뿐만 아니라 오디오 포착 및 모든 시스템 관리, 호출 평형, 로드 평형 기능 등이 제공될 수 있고, 다이얼로그의 드라이버로서 대화식 브라우저만을 사용한다. 다른 방법으로는, 또한 IVR은 다른 서버상에서 실행하는 대화식 엔진 및 또한 다른 서버상에서 실행하는 대화식 브라우저를 가지면서, 오디오 포착 및 로드 평형/시스템 관리만을 실행할 수 있다. 결국, IVR 또는 호출 센터가 오디오 포착, 대화식 엔진 및 하나 또는 매우 제한된 활성 포트(즉, 활성 호출)와 관련된 대화식 브라우저를 소유하는 전화 카드를 각각 갖는 네트워크 소형 워크스테이션(일반적으로 NT 머신)의 팜(farm)에 의해 수행될 가능성이 있다.

도 2에 도시되어 있는 실시예에서, 프롬프트를 재생하고, 음성을 합성하며, 다이얼로그를 발생하고, DTMF 입력 또는 음성 입력을 이해하는 모든 기능은 IVR(101)의 레벨에서 실행되는 것으로 가정한다. CVM(106a)이 없을 때, 시스템 적분기 또는 IVR 벤더의 의무는 (전술한 바와 같이) CVM에 의해 제공되는 것과 유사한 추상 계층을 제공하는 시스템의 확실한 수행을 제공하는 것이다. 이 시스템의 실제의 수행은 IVR(101)의 실제 구조를 결정하는 것이지만, 그러한 구조는 대화식 브라우저(106)의 기능 및 동작에 대하여 무관하다.

대화식 브라우저(106)는 IVR(101)을 당업계에 공지되어 있는 것으로부터 서비스측 상의 패킷 교환 네트워크로의 "올 타임(all-time)" 접속으로 효율적으로 변화시키는 이점이 있다. 또한, 대화식 브라우저를 이용하는 범용 IVR 구조에 의해, 내용 페이지가 동일한 HTTP 서버 상에 있고 정보 액세스용 웹 서버로서 유산 데이터베이스 액세스 SW 계층을 공유하는 동일한 정보 액세스 방법을 사용한다. 환언하면, IVR은 대화식 브라우저를 갖는 HTTP 서버의 특수한 경우를 고려할 수 있다. 장치, 서버 및 내용/서비스 제공자 자격에 따라서, 대화식 브라우저가 클라이언트 및 서버 사이에서 공유되거나 네트워크를 통해 분배되는 서버 상의 클라이언트에 위치될 수 있다.

이하, 도 3을 참조하면, 블록도는 본 발명의 다른 예시적인 실시예에 따른 전화 서비스를 제공하는 대화식 시스템을 도시한다. 이 예에서, 대화식 브라우저(106)는 클라이언트 장치 상에 완전히 위치된다. IVR에 의해 실행되는 모든 처리는 (도 2를 참조하여 전술한 바와 같이) 클라이언트 장치에서 국부적으로 실행되고, HTML 문서의 인출만이 예컨대, HTTP를 통해 내용 서버(105) 상에서 실행된다. 내용 서버(105)는 IVR 어플리케이션과 관련된 자체에서 실행되는 어플리케이션을 갖는 HTTP 서버일 수 있다. 다른 방법으로는, 도 3의 시스템은 WWW를 브라우징하는데 간단히 사용될 수 있다.

예를 들어, 도 3에 도시되어 있는 바와 같이, 클라이언트 장치는 대화식 브라우저(106)(및 CVM(106a)) 및 PC(60)를 통해 전화 서비스를 제공하는 대화식 전화 플랫폼(61)(예컨대, 모뎀, 사운드보드, 스피커폰 및 음성 메일 시스템을



포함)을 갖는 PC(60)(퍼스널 컴퓨터)일 수 있다. PC(60)는 모뎀을 통해(대화식 브라우저(106)를 통해) 전화선(102)을 통하여 팩스로 또는 네트워크(40)(예컨대, 인터넷, 인트라넷)을 통해 내용 서버(105)에 액세스하도록 임의의 다른 원격 액세스 서버(RAS)(70)(예컨대, 인터넷 서비스 제공자(ISP))로 다이얼업 접속을 설정하는데 사용될 수 있다. PC(60)는 또한 전용선(L)을 사용하여 IP 접속을 통해 원격 접속을 통하여 국부/개인 자원 또는 인터넷에 액세스할 수 있다. 브라우저(106)를 갖는 PC(60)는 예컨대, WWW의 대화식 브라우저를 제공하는데 사용될 수 있다. 전술한 바와 같이, 대화식 브라우저(106)는 예컨대, 음성 명령의 인식 및 인터넷을 통하여 압축되어 착신하는 HTTP 서버로부터의 음성 메시지의 변환을 실행한다. 대화식 브라우저(106)는 PC(60) 또는 국부 클라이언트에서 실행하는 임의의 음성 인 에이블 국부 어플리케이션(62)(캘린더 등)에 액세스할 수 있다.

클라이언트 장치는 대화식 브라우저(106)를 국부적으로 실행하기 위해 적절한 구조 및 CPU 전력을 갖는 이동 전화(80)(예컨대, GSM 터미널)일 수 있다. 이동 전화(80)는 음성(인터넷 전화)용 인터넷 프로토콜(또는 WAP와 같은 상이한 프로토콜)을 해석하기 위해 DSP 코드(또는 임의의 다른 유형의 CPU 프로그래밍 언어) 뿐만 아니라 음성용의 압축 알고리즘 및 임의의 다른 어플리케이션 또는 예컨대, 자바의 어떤 파생물로 기록된 음성 처리 프로그램을 다운로드할 수 있다. DSP 코드는 원어일 수 있고, (특히, CVM(106a)이 장치에 설치되어 있는 경우) 전화 상에 프리로드(preload)될 수 있다. 더욱이, 스크린을 갖는 더욱 높은 레벨 이동 전화가 텍스트 능력 또는 디스플레이 능력을 갖는 브라우저를 국부적으로 실행하는 더 높은 레벨을 가질 수 있다. 또한, 새로운 전화에는 더 높은 샘플링 레이트의 A/D 변환기가 설치될 수 있다. 이것은 통신 라인을 통하여 더욱 효율적으로 음성을 전송하고 (상기 통합되어 있는 IBM Docket No. YO999-113P에 설명되어 있는 바와 같은 분배 음성 인식을 위해) 음성 인식 품질을 증가시키도록 셉스트럼 계수(cepstrum coefficient) 압축의 장점을 취할 수 있다.

대화식 브라우저(106)의 기능은 부분적으로는 국부 클라이언트 장치에 위치될 수 있고, 부분적으로는 원격 서버에 위치될 수 있다. 예를 들어, 음향 특징은 클라이언트 장치에 의해 (예컨대, 도 2의 진단부 모듈(107)을 통해) 계산 및 압축되어 처리를 위해 낮은 비트 레이트로 서버에 송신될 수 있다. 또한, 브라우저 처리의 일부, 예를 들어 정규 텍스트의 재생 및 간단한 입력의 수집은 국부적으로 행해질 수 있고, 일부 처리, 예를 들어 스트리트 어드레스 또는 복잡한 입력 또는 ascii 입력을 형식으로 얻는 것은 서버측에서 실행될 수 있다. 더욱이, 브라우저 기능은 클라이언트 장치, 하나 이상의 서버 및 자원의 네트워크간에 분배될 수 있다. 예를 들어, 브라우저는 네트워크 상의 다른 영역에 위치되는 대화식 서비스시스템(예컨대, 음성 인식 엔진, NLU 엔진, TTS)을 가지고 서버, 클라이언트 상에 위치될 수 있거나, 이들 사이에 공유될 수 있다. 브라우징 기능은 네트워킹된 장치(서버, 컴퓨터, PDA 등) 사이에서 공유될 수 있다. 또한, CVM은 통신 프로토콜을 통해 상기 통합되어 있는 IBM Docket No. YO999-111P 및 YO999-113P에 기술되어 있는 것과 유사한 경우를 관리하기 위한 "키" 구성 요소이다. 또한, 도메인 지정 어플리케이션에 있어서, 내용 제공자는 자신의 TTS, NLU 또는 음성 인식 엔진을 직접 제공할 수 있다.

유리하게도, 본 발명의 배치는 HTML 형식의 정보 페이지를 그러한 정보가 (다중 양식 장치에 의해 페이지 또는 페이지의 일부를 디스플레이하기 위한 능력 이외에) 순대화식 대화로 제공될 수 있게 함으로써 제공하는 유산 어플리케이션의 보존을 제공할 수 있다. 각각의 이들 개념은 이하, 도 4a 및 도 4b의 예시적인 실시예를 참조하여 설명한다.

이하, 도 4a를 참조하면, 블록도는 대화식 브라우징 및 IVR 서비스를 제공하는 자원이 하나 이상의 네트워크를 통해 공유되는 분배 시스템을 도시한다. 도 4a의 파선으로 된 원은 예컨대, 소정의 네트워크 또는 인트라넷 또는 인터넷 상의 개별 네트워크(또는 서브네트워크)의 상이한 위치를 나타내고, 이들 위치는 각각 표준 네트워크 프로토콜(예컨대, TCP/IP, HTTP)를 통해 서로 대화하는 대응하는 서버, 호스트를 포함한다. GUI 터미널(300) 및 대화식 브라우저(302a)를 갖는 PC(302)가 GUI 및 대화식 브라우징을 제공하도록 사용될 수 있다(장치(302)는 PSTN을 통해 분배 네트워크에 접속되는 전화, 또는 블루투스를 통해 TCP/IP 네트워크에 접속되는 블루투스 장치, ISDN 또는 케이블 또는 위성 연결을 통해 분배 네트워크에 접속되는 셋톱 박스와 같은 임의의 클라이언트 장치(PC와 다른)일 수 있다). 다른 방법으로는, PC(302)가 IP 접속을 통해 대화식 브라우저를 갖는 원격 서버(303)에 액세스할 수 있다. 유사하게, 사용자는

전화(301)(예컨대, 정류, 셀룰러 또는 스마트 폰)을 통해 IVR 서버/PC(302)로 다이얼할 수 있고, 국부 브라우저(302a)를 통해 또는 원격 서버(303) 및 대화식 브라우저(303a)를 통해 대화식 브라우징 또는 IVR을 실행할 수 있다. 다른 방법으로는, 전술한 바와 같이, 전화(301), PC/IVR 서버(302) 및 원격 서버(303)가 각각의 구조 및 처리 전력 능력에 따라서 대화식 브라우저의 기능을 공유할 수 있다. 또한, 대화식 브라우저(303a)는 소정의 네트워크 또는 서브네트워크를 통해 분배되는 하나 이상의 서버(304)간에 공유될 수 있다. 다른 방법으로는, 각각의 원격 서버(304)는 브라우저(304a)의 하나 이상의 특수한 기능을 지원할 수 있다. 각각의 대화식 브라우저 기능에 따라서, PC/IVR 서버(302) 또는 원격 서버(303, 304)는 각각 TCP/IP, HTTP, API(전술한 통신 프로토콜을 사용하는 CVM 예시를 위한)를 통해 네트워크 상에 분배된 하나 이상의 대화식 서브시스템(309)에 (필요에 따라) 호출할 수 있다. 또한, 임의의 요구된 대화식 인수(308)는 대화식 브라우저(302a, 303a 및/또는 304a) 또는 네트워크 상의 대화식 서브시스템(309)에 의해 사용하기 위해 액세스될 수 있다.

예시적인 시스템은 HTML 문서 및 다른 어플리케이션을, 대화식 브라우저(302a, 303a, 304a)가 브라우저가 실행되는 장치의 부적절한 처리 전력으로 인해 그러한 변환을 실행할 수 없는 환경에서 CML로 변환하기 위해 네트워크 상에 액세스될 수 있는 트랜스코더(305a)를 갖는 대화식 프록시 서버(305)를 더 포함한다. 또한, 대화식 프록시 및 트랜스코더(305a)는 대화식 브라우저(302a, 303a 및/또는 304a)가 그러한 변환을 실행하기 위한 어플리케이션에 관한 특정 특허 정보(내용 제공자에 의해서만 알려짐)의 부족 시에, HTML 문서의 변환을 실행하도록 내용 서비스 제공자와 결합될 수 있다. 도 4a에 도시되어 있는 바와 같이, HTML 문서는 특허 정보를 HTML 문서로 변환하도록 SQL 또는 특허 API를 통해 데이터베이스(307)와 대화하는 통상의 트랜스코더(306)를 사용하여 내용 제공자의 웹 서버 어플리케이션을 통해 발생된다. 구성 요소(306 및 305/305a) 사이의 통신은 대화식 트랜스코딩/프록시 작업을 보조하기 위해 XML 또는 메타 정보의 전송을 수반할 수 있다. 또한, 구조 골격이 상기 통합되어 있는 출원 Y0999-114에 기재되어 있는 바와 같이 사용될 수 있다.

도 4a는 트랜잭션이 현재의 웹 포털과 유사한 HTML 또는 CML 페이지 또는 웹 사이트의 탐색을 가능하게 하는 CML 서버인 음성 포털을 도시한다. 음성 포털은 원하는 사이트 및 페이지를 검색할 수 있다. 이들 CML 페이지는 대화식 브라우저에 직접 표시된다. 다른 표시 언어(예컨대, HTML)로 검색되는 페이지는 브라우저에 송신되기 전에 트랜스코딩된다. 음성 및 웹 포털(즉, HTML 또는 WML(무선 마크업 언어) 기반) 사이의 주요 차이점은 요구된 페이지를 다운로드하여 이 페이지를 브라우저에 전송하기 전에 적절하게 이 페이지를 트랜스코딩하는 것이다. 트랜스코더를 갖는 클라이언트 장치(또는 브라우저)에 있어서, 페이지는 다른 포맷(HTML 및 Vor HTML 기반 구조 골격)으로 브라우저에 전송될 수 있고, 중간 내부 언어로서 브라우저에 의해 CML로 플라이 상에서 트랜스코딩될 수 있다. 음성 포털 서비스는 완전한 이익이 있는 트랜잭션에 근거하여, 언급된 비즈니스로부터의 이익의 백분율에 근거하여, 검색 엔진에서 제공되는 서비스 또는 검색 엔진의 결과로 높은 순위의 서비스를 갖거나, 자동화 방법과 비교하여 (수동 준비되거나 재검토되는) 품질 CML 페이지에 대해 가입 기준으로, 캐리어/무선 제공자에 의해 지불되고, 언급된 내용 제공자에 의해 (히트 식으로) 지불되는 가입 기준일 수 있다.

이하, 도 4b를 참조하면, 도면은 유산 기업 소프트웨어 어플리케이션과 관련하여 대화식 브라우징 또는 IVR 서비스를 제공하는 본 발명의 예시적인 실시예를 도시한다. 도 4b의 시스템은 도 4b가 소정의 기업 네트워크 상에 종래의 3 계층(분배) 기업 어플리케이션의 대화식 브라우징을 특히 나타내는 것을 제외하고 전술한 시스템(도 4a)과 유사하다. 3 계층 어플리케이션은 HTML 기반 문서를 생성 및 표시하도록 프로그래밍을 포함하는 전단부 터미널/표시 서버(310)(계층 1), 중간 계층 비즈니스 논리 서버(311) 또는 어플리케이션 서버(비즈니스 어플리케이션을 포함하고, 예컨대, LAN내의 워크스테이션으로부터의 클라이언트 요구를 서비스함) 및 유산 데이터베이스 및 트랜잭션 관리(데이터 제어) 어플리케이션을 포함하는 후단부 또는 제3 계층(312, 313)을 포함한다. 도시되어 있는 바와 같이, 종래의 표시는 GUI(312) 전화(IVR)(315) 및 3270 유형 스크린 다이얼로그(3270 터미널은 메인 프레임 및 CISC 시스템에 의해 일반적으로 지원되는 유형의 터미널 예컨대, 대부분 ascii 문자 기반 스크린인 ATM에서 볼 수 있는 통상의 녹색 터미널이다)를 통해 달성된다. 형식 채움과 같은 직송식 HTML 다이얼로그 페이지(HTML 또는 CGI) 또는 ATM 다이얼로그(

및 다른 3270 유형의 스크린 다이얼로그) 또는 다른 의사 터미널(CT(100), 팜 파일럿 스크린 등)이 예를 들어, 그러한 트랜잭션의 비즈니스 논리가 일반적으로 간단한 다이얼로그에 의해 구동되어 제한된 크기의 정보 또는 선택이 사용자에게 1회 표시되기 때문에, 프록시 서버(305)의 트랜스코더(305a)를 통해 CML로 쉽게 변환될 수 있다. 선택 또는 메뉴는 상기 통합되어 있는 IBM Docket No. YO999-114P에 기재되어 있는 바와 같은 골격을 통해 전송될 수 있다. 또한, 트랜스코딩은 다른 특허 API 또는 스크린 맵(예컨대, 스크린 스크래퍼로 칭하는 맵핑 프로그램에 의해 통상 실행되는 HTML 페이지로의 3270 스크린 또는 다른 스크린의 맵)으로 쉽게 확장될 수 있다. 특히, 프로그램이 현재 스크린 맵을 HTML 페이지로 트랜스코딩하기 위해 존재하고, 스크린 맵이 비즈니스 논리를 구동시키는 간단한 다이얼로그로 제한되기 때문에, 결과적인 HTML 페이지의 변환은 대화식 프록시의 사소한 경우이다.

유리하게도, 각각의 전술한 예시적인 시스템 구조는 HTML 및 다른 XML/HTTP 프로토콜 상에 현재 제공되는 내용의 임의의 내용의 재설계를 필요함이 없이 동일한 프로토콜을 통하여 트랜잭션 및 정보 서비스의 동일한뱅크로의 순 음성 인터페이스(대화식 브라우저), 순 GUI 및 음성 및 GUI의 조합을 통한 범용 액세스를 가능하게 한다.

#### CML을 이용하는 대화식 어플리케이션

예시적인 실시예의 이하의 설명은 본 발명에 따른 CML(대화식 마크업 언어)을 사용하는 네트워크 기반 대화식 어플리케이션을 설명한다. 여기에 사용되는 용어 "대화식 어플리케이션"은 음성 입/출력을 통해 사용자와 대화하는 어플리케이션을 칭한다. 여기에 사용되는 용어 "네트워크 기반 대화식 어플리케이션"은 구두 출력 및 입력 즉, CML의 페이지를 한정하는 변환의 구성 요소 중 하나가 네트워크 상에서 또는 국부적으로 얻어질 수 있는 것을 칭한다. 실제로, GUI 어플리케이션이 HTML 마크업 언어를 사용하여 기록된 한 세트의 상호 연결 페이지를 사용하여 형성될 수 있는 것과 같이, 대화식 시스템은 대화식 마크업 언어 - CML로 기록되는 한 세트의 상호 연결 페이지를 사용하여 형성될 수 있다. 더욱이, HTML이 GUI 동작을 다른 페이지를 방문하거나 서버와 통신하는 것과 같은 어플리케이션 동작으로 해석하는 한 세트의 메커니즘을 제공하는 것과 같이, CML은 유사하게 구두 입력을 어플리케이션 동작으로 해석하는 메커니즘을 제공하도록 수행될 수 있다. 이 문맥에서의 용어 어플리케이션은 텍스트, 링크 및 형식을 형식을 처리하는 서버 상의 코드(예컨대, 서브릿 또는 CGI 스크립트)와 함께 포함하는 한 세트의 관련 HTML 페이지가 어플리케이션을 구성하는 넓은 의미로 이해되어야 한다. 또한, 대화 기본 등급 또는 다른 다이얼로그 구성 성분과 같은) 절차가 자바 또는 액티브X 구성 성분을 통해 로드될 수 있는 방법과 유사하게, 대화식 애플릿 및 기본 등급이 절차적인 대화 동작(CVM의 특징)을 실행하도록 로드/다운로드될 수 있다.

이하, 도 5를 참조하면, 도면은 본 발명에 따른 CML에 의해 인에이블되는 대화식 어플리케이션의 여러 가지 등급을 포함하는 예시적인 네트워크 기반 대화식 시스템을 도시한다. 예시적인 시스템은 대화식 브라우저(501)를 갖는 클라이언트 장치(500)를 포함한다. 클라이언트 장치(500)는 예컨대, 데스크탑 PC, PDA, 자동차용 컴퓨터, 스마트 전화, 또는 "덤(dumb)" 전화(용어 "덤" 전화는 여기에서 긴 코드의 단부에서 마이크로폰으로서 기능하고 전화선의 타단부에서 브라우저에 접속될 필요가 있는 장치로서 언급된다)일 수 있다. 클라이언트 장치(500)는 또한 대화식 브라우저(501)와 통신하는데 CML을 사용하여 클라이언트(500)에서 실행하는 하나 이상의 음성 인에이블 국부 어플리케이션(502)을 포함할 수 있다. 예를 들어, 국부 어플리케이션은 "음성 네비게이션 어플리케이션"이 대화식 드라이빙 방향을 제공하기 위해 컴퓨터 맵핑 소프트웨어 및 GPS(전세계 측위 시스템)와 대화하는 차량 네비게이션 어플리케이션일 수 있다. 유사하게, 종래의 하위 OS에 의해 국부적으로 제공되는 다른 어플리케이션 및 서비스, 또는 다른 어플리케이션이 그러한 서비스를 제공하기 위한 하위 CVM이 없을 때, 서비스(명확성, 문맥 액세스 등과 같은) 및 다이얼로그 모듈을 브라우저에 제공하는데 사용될 수 있다.

클라이언트 장치(500)는 대화식 UI를 장치(500)의 사용자에게 제공하기 위해 대화식 브라우저(501)에 의해 처리되는, 클라이언트 장치(500)에 CML 페이지를 송신하도록 표준 네트워크 프로토콜(예컨대, HTTP, TCP/IP)을 사용하여 네트워크(503)(예컨대, 인터넷)상의 복수의 서버 시스템(S1, S2, S3)중 하나에 액세스할 수 있다. 예를 들어, 클

라이언트 장치(500)는 전술한 바와 같이, HTML을 해석하고 관련 정보를 추출하며 CML 정보를 발생할 수 있는 트랜스코딩 프로세스를 통해 기존의 HTML 정보에 액세스하도록 서버(S1)에 접속할 수 있다. 또한, 클라이언트 장치(500)는 CGI 어플리케이션을 통해 노트 데이터(예컨대, 노트 e-메일)에 액세스하기 위해 로터스 도미노 서버와 같은 특수한 웹 서버 어플리케이션(S2)에 접속할 수 있다. 특히, 도미노 서버는 CML을 발생하도록 구성될 수 있고, HTTP를 사용하여 CML을 송신한다. 다른 예에서, 클라이언트 장치(500)는 기업의 유산 데이터베이스에 액세스하기 위해 CGI 어플리케이션 또는 자바 서브릿을 사용하여 웹 서버 어플리케이션(S3)에 접속할 수 있으며, 여기에서 웹 어플리케이션은 CML로 정보를 발생 및 송신한다.

이것은 다른 흥미있는 어플리케이션 예: CML 기반 범용 메시징 시스템을 유도한다. 특히, 종래 및 현재 사용 가능한 메일 및 메시징 시스템에 ISP, ASP(어플리케이션 서비스 제공자) 및 웹 포털에 의해 HTML이 제공될 수 있는 것과 동일한 방법으로, 음성 포털 또는 음성 웹 서비스 제공자가 e-메일 액세스 및 처리 및 범용 메시징을 제공하기 위해 CML을 사용할 수 있다. 범용 메시징은 이제 음성을 통해 상이한 메시징 서비스 및 프로그램 호출, e-메일, 음성 메일 및 음성으로 처리하는 호출에 액세스할 수 있다는 것을 의미한다. 인터페이스가 대화식 브라우저를 통해 CML에 형성되기 때문에, 사용자는 자신의 메시지를 자동으로 처리할 수 있다. 유사하게, 호출 전송 및 처리는 호출자 정보 및 가능하게는 호출 토픽(topic)을 고지하고 옵션의 메뉴를 제공하는 CML 인터페이스의 조합으로 보일 수 있다. 사용자는 동작(호출을 전송, 저장, 포착)을 선택할 수 있다. 각각의 이들 동작은 절차적인 호출로서 호출 처리 유닛에 처리되고, 호출은 그에 따라 처리된다(예컨대, 사용자에게 전송된다). 도 5에 도시되어 있는 것과 같은 순 음성 대화식 브라우저를 이용하면, 음성 및 오디오만이 사용자에게 제공되는 출력이다. 사용자가 원할 때, 페이지가 서비스 제공자로부터 (CML로) 다운로드된다. CML은 사용자에게 제공될 대화식 UI를 나타내는데 사용된다. TTS 마크업 정보를 사용하면, 대화식 브라우저는 사용자에게 페이지를 판독하고, 링크에 따르고, 형식을 채우거나 브라우저를 명령하도록 (CML 페이지의 음성 인식부로부터의 표지를 사용하여) 음성 인식기에 의해 디코딩될 사용자로부터의 명령을 청취한다. 비삽입 어플리케이션에 있어서, 대화식 브라우저는 바람직하게는 큰 어휘 연속 음성 인식 시스템(예컨대, 음성을 통한 IBM)과, 1999년 9월에 헝가리 부다페스트에서 열린 Proc. Eurospeech 1999에서 K. A. Kishore 등의 "Free-flow dialog Management Using Forms" 라는 명칭의 참조 문헌 및 1999년 9월에 헝가리 부다페스트에서 열린 Proc. Eurospeech 99에서 Davies 등의 "The IBM Conversational Telephony System For Financial Applications" 라는 명칭의 참조 문헌에 기재되어 있는 바와 같은 동적 문법 편집 및 NL 처리를 포함하는 "JSAPI(자바 음성)를 사용하는 프리 텍스트 규칙 기반 TTS"를 이용한다.

유리하게도, 서버 기반 CGI 스크립트와 자바 서브릿 및 국부 플러그인과 같은 상기 자격은 대화식 어플리케이션에 참여하도록 어플리케이션 지정 코드용 메카니즘을 제공할 수 있다. 어플리케이션 코드 및 대화식 브라우저(501) 사이의 인터페이스는 요구 및 마크업 데이터 스트림(HTML 또는 CML)에 대해 URL의 형식 또는 속성값 n-uples의 세트를 취할 수 있다. URL 대신에, 국부 또는 분배 서비스에 액세스하기 위한 다른 어드레싱안(예컨대, 소켓 및 포트)이 사용될 수 있다.

각각의 선행 IVR 및 대화식 브라우징 어플리케이션은 다른 방법으로 API 및 원격 호출(즉, 대화식 브라우저 및 CML을 수행하지 않음)을 통해 통신하는 어플리케이션의 어플리케이션 코드에 의해 형성될 수 있다. 특히, 그러한 어플리케이션은 자바 음성 API(JSAPI)와 같은 음성 API에 직접 기록될 수 있다. 그러나, 분배 어플리케이션에 있어서, 현재 사용 가능한 엔진 API(SAPI, JSAPI)는 네트워크를 고려하여 설계되지는 않는다. 따라서, 대화식 엔진 API의 상부에 직접 기록되는 네트워킹/분배 음성 어플리케이션은 항상 분배 기능의 재수행을 요구한다. 그러나, CVM 시스템(IBM Docket No. YO999-111P)의 상부에 기록되는 어플리케이션에 대하여, 현재의 엔진 API의 이들 제한은 CVM 셀에 의해 제공되는 서비스 및 동작을 통해 엔진 API의 대부분의 복잡성을 숨기는 통신 스택을 통해 통신되는 대화식 어플리케이션 API를 통해 가상적으로 소거된다. 실제로, 복잡한 다이얼로그 작업은 절차적인 수행을 통해 효율적으로 실행될 수 있다. 그러한 절차는 그러한 복잡한 작업에 대한 매크로의 간단한 사용을 가능하게 한다. 또한, 절차는 XML과 같은 해석된 서술 언어보다 더욱 효율적인 것으로 기대된다. 사용의 문맥(효율성 대 프로그래밍, 플랫폼간의 휴대성 등의 중

요성)에 좌우될 수 있으나, 양 대안(서술 및 절차)이 사용될 수 있다.

한편, CML 및 대화식 브라우저는 유리하게도 분배 대화식 시스템이 예컨대, HTML을 사용하여 이미 개발된 유산 어플리케이션의 장점을 취하여 수행될 수 있게 한다. 현재 사용 가능한 음성 API와 달리, CML은 네트워킹된 대화식 어플리케이션을 쉽게 지원할 수 있다. 대화식 브라우저는 다중 위치로부터 발신하는 CML의 스트림을 해설할 수 있고, 그것에 의해 데스크탑 윈도우 시스템에 의해 또는 가상 브라우저에 의해 제공되는 동일한 종류의 이음매없는 경험을 사용자에게 제공한다. 예를 들어, 항공 티켓 구매를 경합하는 동안, 사용자는 트랜잭션을 일시 정지시킬 수 있고, 예금 잔고를 체크하기 위해 상이한 서버 상의 금융 어플리케이션과 대화할 수 있다. 이것은 브라우저와의 다이얼로그의 문맥을 유지 및 송신/저장하고, 가능하다면 서버측에서 처리하는 적절한 메카니즘을 갖는 전술한 개념과 관련된다. 더욱이, CML은 (HTML이 시청 어플리케이션의 생성을 간략하게 하는 방법과 유사하게) 대화식 어플리케이션의 구성을 간략하게 하는 높은 레벨 토큰을 제공한다. 대화식 마크업 언어는 또한 요구되는 음성의 전문 지식의 양을 감소시킨다. 또한, 전술한 바와 같이, 현재 사용 가능한 음성 API를 사용하여 이중 플랫폼에 분배 어플리케이션을 생성하는 것은 어렵다. WW를 통해 이중 플랫폼에 분배 GUI 어플리케이션의 대규모 배치는 CML 데이터 스트림 방법의 가능성을 나타낸다.

도 6을 참조하면, 블록도는 본 발명의 일 실시예에 따라 CML을 사용하는 통상의 분배 대화식 브라우징 시스템을 도시한다. 통상적으로, 분배 대화식 브라우징 시스템은 정보 서버(600), 표시 서버(601), 음성 서버(602), 음성 모델(603) 및 대화식 브라우저(604)를 포함한다. 정보 서버(600)는 예컨대, 기업 데이터베이스, 뉴스와이어 피드(newswire feed) 및/또는 WWW 서버를 나타낸다. 정보 서버(600)는 장치 독립적인 방식으로 어플리케이션 지정 정보를 나타내는 "정보 포맷"으로 데이터를 제공한다. 이러한 데이터 포맷의 예는 데이터 액세스용 API(예컨대, 노트 DB API, SQL) 및 어플리케이션 지정 데이터 스트림(뉴스 와이어 피드, XML 내용 태그로 마크업된 정보)를 포함한다.

표시 서버(601)는 API 또는 정보 서버(600)에 의해 한정되는 프로토콜을 사용하여 정보 서버(600)로부터 정보를 검색한다. 표시 서버(601)는 검색된 "정보 포맷"을 사용자에게 표시하기 위한 "표시 포맷"으로 트랜스코드한다. "표시 포맷"은 본 발명에 따르면 (이하 더욱 상세히 설명되는 바와 같이) CML인 사용자에게 표시하기 위한 장치 지정 어플리케이션 독립적인 방법으로 정보를 표시한다. 종래의 "표시 포맷"은 예컨대, 윈도우즈 GUI, API 및 HTML, HDM L(휴대용 장치 마크업 언어) 및 WAP(무선 어플리케이션 프로토콜)과 같은 데이터스트림을 포함한다. 표시 서버(601)는 예컨대, HTML 서버, CGI 스크립트 및 자바 서브릿, HTML 서버로서의 도미노, 및 정보 서버(600)로부터 수신된 정보를 CML로 트랜스코딩하기 위해 구성된 노트 클라이언트일 수 있다. 표시 서버(601)에 의해 발생된 CML은 대화식 브라우저(604)를 통해 사용자에게 표시될 대화식 UI를 나타내는데 사용되어, 구두 입력을 어플리케이션 동작으로 해석하는 메카니즘을 제공한다.

음성 서버(602)는 음성 인식 및 분석을 행하는 엔진 및 다른 대화식 엔진을 포함한다. 음성 서버(602)는 음성 특유의 문제점에 어드레스하고: 음성 인식 프로세스는 클라이언트에게 송신 불가능한 대형 어플리케이션 지정 데이터 세트(즉, 음성 모델(603))를 요구할 수 있다. 이것은 음성 인식 프로세스가 언어 모델이 (송신 대역폭의 의미에서) 표시 서버(601) 근처에 있는 위치에서 실행되어야 하는 것을 의미한다. HTTP, XML, VoIP 프로토콜과 같은 표준 네트워크 프로토콜 뿐만 아니라 상기 통합되어 있는 YO999-113P에 기재되어 있는 대화식 음성 코딩 프로토콜은 대화식 브라우저(604)로부터 음성 서버(602)로 오디오를 송신하고 분석 결과를 복귀시키는데 사용될 수 있을 뿐만 아니라 음성 서버(602)로부터 대화식 브라우저(604)에 기록되거나 합성된 음성과 같은 오디오를 송신하는데 사용될 수 있다.

이하, 도 7을 참조하면, 도면은 본 발명의 일 실시예에 따라 CML을 사용하는 예시적인 분배 대화식 시스템을 도시한다.

특히, 예시적인 실시예에는 대화식 브라우저(604) 및 마이크로폰/스피커(605)를 갖는 음성 전용 클라이언트 장치(606)를 설명한다. 클라이언트 장치(606)는 예를 들어, 예컨대, 무선 링크에 의해 네트워크(607)(예컨대, 인터넷)에 접속되는 음성 전용 PDA(개인 휴대용 정보 단말기) 또는 PVA(개인 차량용 정보 단말기) 클라이언트 장치를 나타낸다. 정보 서버(600), 표시 서버(601) 및 (음성 모델을 갖는) 음성 서버(602)는 함께 CML을 사용하여 클라이언트 장치에 어플리케이션(예컨대, 날씨 정보)을 제공한다. 표시 서버(601), 음성 서버(602) 및 음성 모델(603) 둘레의 파선은 이들 구성 요소가 단단히 결합되어 음성 서버(602)가 어플리케이션에 지정된 음성 모델을 가지는 것을 나타낸다. 또한, 표시 서버는 HTML을 사용하여 GUI를 다른 브라우저(608)에 제공한다. 대화식 브라우저(604)는 네트워크(607) 상의 다른 대화식 어플리케이션에 액세스할 수 있다.

이하, 도 8을 참조하면, 도면은 클라이언트 장치(606)가 음성 및 GUI 클라이언트인 다른 예시적인 분배 대화식 시스템을 도시한다. 클라이언트 장치(606)는 디스플레이 스크린(609), 마이크로폰 스피커(605), 및 음성, GUI 및 이들의 조합을 사용하여 사용자에게 정보를 표시하는 대화식 GUI 브라우저(604a)를 포함한다. 이 실시예에서, 표시 서버(601)는 클라이언트(606)에 의한 요구에 따라 HTML 및 CML 문서를 모두 제공할 수 있다. 또한, 이 시스템은 HTML 마크업에 삽입될 수 있는 CML 마크업을 삽입하고, GUI 인터페이스와 나란히 대화식 인터페이스를 제공하기 위해 표준 HTML 브라우저를 (공급되는 브라우저 확장 메커니즘을 사용하여) 표준 브라우저와 나란히 대화식 브라우저를 실행하도록 확장함으로써 수행될 수 있다. GUI 및 음성 UI 사이의 통합 및 조정을 위한 시스템 및 방법은 1999년 4월 7일에 출원되어 공동 양수되어 참고로 본 명세서에 통합되어 있는 "Multi-Modal Shell" 이란 명칭의 미국 가출원 60/128,081호에 기술되어 있다.

이하, 도 9를 참조하면, 분배 전화를 사용하는 또 다른 분배 대화식 시스템이 도시되어 있다. 이 예시적인 실시예에서, 사용자는 사용자 액세스를 네트워크(607)(예컨대, 인터넷)를 통해 음성 어플리케이션의 선택에 제공하는 대화식 브라우저(604)에 접속될 클라이언트(606)(PC 등)의 전화 플랫폼(610)으로 다이얼할 수 있다. 다른 방법으로는, 사용자는 다이얼 입력을 통해서 또는 또 다른 기존의 접속 장치(LAN, 블루투스, DSL 등)를 통해서 음성 서버에 접속할 수 있다. 이 경우에, 초기에 제공되는 대화식 코딩 알고리즘은 데이터를 브라우저에 보내기 위하여 사용된다. 클라이언트 "장치"는 현재 전화 플랫폼(610)상에서 실행되는 대화식 브라우저(604)에 반드시 결합되는 전화 핸드셋이라는 점에 유의하라.

도 10의 예시적인 실시예에서, 사용자는 소정의 전화 번호로 다이얼링하여 하나의 어플리케이션에 액세스한다. 이 시스템은 도 9의 분배 전화 시스템의 간략화된 버전인데, 여기서 대화식 브라우저(604) 및 어플리케이션은 동일하다. 그러나, 이 실시예에서, 대화식 브라우저(604)는 네트워크에 접속되는 것이 아니라 단지 특정된 어플리케이션에만 액세스할 수 있다. 이 실시예에서 파선(611)은 표시 서버(601), 음성 서버(602)(및 모델(603)), 대화식 브라우저(604) 및 전화 플랫폼(610)이 통합된 서비스로서 모두 동작된다는 점을 강조한다. 이것은 반드시 통상적인 전화 어플리케이션인데, 상기 대화식 브라우저(604)는 다이얼로그 엔진으로서 작용한다.

대화식 마크업 언어 : CML

이하에서 전술한 대화식 브라우징 어플리케이션에서 사용될 수 있는 본 실시예의 CML에 대해 설명한다. 본 실시예에서, CML은 확장 가능한 마크업 언어(XML)의 어플리케이션이다. XML은 HTML 표준을 유지하는 동일한 본문인 W3C로 규정된 HTML - 스타일 마크업 언어에 대한 대중적인 웹 표준이다. XML은 HTML를 일반화한 것인데, 이 XML은 HTML로부터 다수의 설계 포인트를 차용한 것이다. 보다 일반적으로 말하면, XML은 구조화된 무선 및 데이터에 대한 선언적인 통합 포맷이다. CML에 대한 기본으로서 XML을 사용하면 CML이 HTML에 쉽게 삽입되거나 HTML과 결합되어 멀티 모델 어플리케이션(예를 들어 음성 및 GUI)을 생성하는데 유용하게 된다. 이것은 또한 JSML(Java Synthes

is Markup Language), 또다른 XML 어플리케이션이 CML에 삽입되도록 하고 CML을 위한 음성 합성 마크업 언어로 사용되도록 한다. XML의 표준화 및 대중성은 CML을 발생시키기 위한 서버 코드용 프로그래밍 툴(programming tool) 뿐만 아니라 CML 페이지 생성을 위한 인증 툴(authoring tool)이 표준이 되도록 하여 본원에 서술된 대화식 브라우저용 공개 구조를 제공하는데 유용하다.

XML은 페이지 또는 파일의 내용을 서술하기 위한 마크업 심벌을 포함한다. XML 엔티티는 다음 형태의 태그 쌍들을 포함한다.

< tagattr1= " val2"attr2= " val2" >arbitrary text< tag> .

XML의 확장성은 XML 어플리케이션(이 경우에 CML)의 마크업 심벌이 제한되지 않고 자체 정의(즉, 프로그래머는 자신의 태그 명칭, 속성 명칭 및 값 세트를 규정하는데 자유롭다)되어 있다는 점으로부터 발생된다.

본 발명의 일 실시예에서, CML은 JAVA로 기록되고 대화식 브라우저는 국부적인 음성 인식 및 문장 분석을 위하여 "당신이 듣고 있는 것은 당신이 말할 수 있다" 라는 음성 입력 스타일을 사용한다. 다른 실시예에서, CML은 NLU 음성 입력 및분배 클라이언트 - 서버 음성 어플리케이션을 위하여 확장되는데, 여기서 디지털화된 음성은 예를 들어 HTTP를 통해서 클라이언트로부터 음성 서버로 전송되고 문장 분석된 결과는 CML(전술한 바와 같음)을 사용하여 복귀된다. 또 다시, HTTP 또는 그외 다른 종래의 프로토콜을 사용하는 것 이외에, 음성은 상기 본원에 포함된 특허 출원 IBM Doc ket Nos. YO999 - 111P 및 YO999 - 113P에 서술된 바와 같은 종래의 통신 프로토콜을 통해서 음성 서버로 전송될 수 있다. 이들 실시예는 후술할 것이다.

일반적으로, CML은 태그의 세트로서 규정되는 것이 바람직한데, 여기서 1차 CML 구성 요소는 < page> , &lt; body> , < menu> 및 < form> 이다. 일반적으로, " 페이지" 구성 요소는 다른 CML 구성 요소를 그룹화하여 CML 문서용 최상의 레벨 구성 요소로서 작용한다. < body> 구성 요소는 대화식 브라우저에 의해 말하여진 출력을 규정한다. < menu> 구성 요소는 선택 목록을 사용자에게 제공하여 각각의 선택을 목표, 예를 들어 사용자가 이 선택을 택한 경우 방문할 CML 구성 요소를 식별하는 URL과 관계시킨다. < form> 구성 요소는 사용자로 하여금 하나 이상의 정보 조각(a piece of information)을 제공하도록 하는데, 여기서 각각의 정보 조각의 내용은 문법(grammar)으로 서술될 수 있다. < form> 구성 요소는 또한 사용자가 이 형식을 완성하였을 때 방문할 URL을 규정한다. 이 페이지 " 구성 요소 방문" (또는 CML 구성 요소 방문)은 통상적으로 사용자 구어 입력에 응답(비록 다른 종류의 사용자 입력 또는 어떤 다른 종류의 비동기 이벤트에 응답할 수 있지만)하여 대화식 브라우저에 의해 취해지는 동작과 관계한다. CML 구성 요소의 방문은 방문된 CML 구성 요소의 타입 및 내용에 따라서 대화식 브라우저로 하여금 구어의 출력을 발생시키도록 한다. 예를 들어, < body> 구성 요소는 판독되며, < menu> 구성 요소의 선택은 목록을 작성하는 등등이다. 이하에서 각각의 구성 요소에 대하여 서술하고 다이얼로그 이슈의 섹션에서 보다 자세하게 논의된 바와 같이 구성 요소 방문은 또한 브라우저가 사용자로부터 수용할 구어 응답 세트에 영향을 미칠 수 있다.

이하에서 본 실시예에 따른 CML 문서 또는 파일을 포함하는 속성 및 태그(구성 요소)에 대해서 보다 상세하게 설명한다.

페이지 구성 요소: < page> 구성 요소는 하나 이상이 관계된 네스팅된 CML 페이지 또는 유닛(유닛은 < body> , < menu> 또는 < form> 구성 요소이다)을 포함하는 최상 레벨의 CML 문서 구성 요소이고 다음과 같은 구조로 되는 것이 바람직하다.

< page ttl = " seconds" > (본문, 메뉴 및/또는 형식 유닛) 또는 네스팅된 페이지< /page>

이 속성(ttl)(" time-to-live")은 CML 페이지가 캐시에 기억되는 세컨드 수를 규정하는데, 여기서 0의 값은 페이지의 캐시를 방지한다. < page> 구성 요소 그 자체는 대화식 브라우저 어플리케이션 및 사용자간의 대화에 기여(구어의 출력 또는 입력이 존재하지 않음)하지 않지만 다른 CML 구성 요소(예를 들어, 메뉴, 본문 및 형식 구성 요소)를 그룹화하기 위한 컨테이너로서 작용한다. < page> 구성 요소는 또한 네스팅된 페이지를 포함할 수 있다. (네스팅된) 컨테이너로서 < page> 구성 요소의 기능이 후술될 것이다. < page> 구성 요소의 방문은 페이지에서 제1 유닛을 방문하는 것과 동등하다. 네스팅된 페이지는 또한 대화식 다이얼로그 오브젝트 또는 파운데이션 클래스(foundation classes)를 포함할 수 있다. 이것은 절차에 따라서 또는 어떤 등가의 CML 페이지로 구현될 수 있다. 페이지가 재로딩될 때, 쿠키(cookie) 또는 적절한 프로토콜/서비스/API는 사전 컨텍스트에 제공된다.

메뉴 구성 요소: < menu> 구성 요소는 선택 세트를 사용자에게 제공한다는 점에서 하이퍼텍스트 링크 및 메뉴가 HTML에 서비스하는 기능과 유사한 CML 기능을 서비스한다. 일반적으로, < menu> 구성 요소는 엔티티가 방문될 때 말하여지는 프롬프트 메뉴 및 서두 메시지와, 대응하는 가능한 응답 세트 및 각각의 선택에 대해서 사용자가 이 선택을 택한 경우 방문할 예를 들어 URL을 규정한다.

< menu>

서두 메시지 텍스트

< choice target =" URL1">prompt text 1< /choice>

< choice target =" URL2">prompt text 2< /choice>

< /menu>

< menu> 구성 요소가 방문될 때, 이것의 타이틀 텍스트(서두 텍스트)는 임의의 포함된 < choice> 구성 요소보다 앞서 말하여진다. < menu> 의 타이틀 텍스트에 정합하는 문법은 < menu> 가 로딩될 때 활성화된 후 활성 상태로 유지한다. 사용자가 < menu> 구성 요소의 타이틀 텍스트에 정합하는 워드 또는 페이지를 말할때, 이 < menu> 구성 요소가 방문된다.

각각의 < choice> 은 사용자가 이 선택을 택한 경우 대화식 브라우저가 방문하는 목표 URL을 갖는다.

< choice target = " target" >prompt text< /choice>

목표는 임의의 유효한 URL(HTTP, 파일, FTP 및 로컬:프로토콜(로컬:프로토콜에 대해서 이하의 설명을 참조하라)을 규정하는 상대적인 URLs 및 URLs) 또는 URL 과 더불어 기준(표준 URL # 기준 신택스를 사용)일 수 있다. 임의의 다른 어드레스, 방식 및 프로토콜이 사용될 수 있다. 이 기준(즉, #후의 URL 파트)은 그 값이 기준 스테어링인 명칭 속성을 갖는 문서내의 유닛을 명명하는 스테어링(string)이다. 기준이 없는 URL은 문서내의 제1 유닛에 관계하는 것으로 간주된다. 상대적인 URL은 포함하는 문서의 URL과 관계하여 해석된다.

또한, " 목표" 는 소켓 어드레스(IP 어드레스 및 포트 ID)와 같은 또다른 어드레스 일 수 있다. 이와 같은 경우에, 본원에 포함된 상기 특허 출원 IBM Docket No. YO999 - 111P 및 YO999 - 113P에 서술된 대화식 프로토콜과 같은 다른 IP 프로토콜이 수행될 수 있다. 종래 프로토콜이 아닌 특정한 프로토콜이 사용되는 상황에서, 부가적인 논의가 다음과 같은 통신을 활성화시키기 위하여 사용될 수 있다.

< choice target = " target" Protocol = " protocolidentifier" > < /choice>



이것은 컨텍스트 및 메타 정보를 보내는 메카니즘을 제공한다. 이것은 또한 시스템 호출을 CVM 서비스에 제공하거나 CVM의 부재시에 호출을 기본적인 OS의 최상부상의 어플리케이션으로서 실행되는 등가의 서비스에 또는 브라우저 플랫폼 자체내에 제공하는 방식을 제공한다. 브라우저가 프로토콜을 지원하는 경우, 식별자는 적절한 통신을 활성화시키는데 충분하게 된다. 이것은 특히 CVM(전송되고 IBM Docket No. YO999-111P에 서술된 바와같음)의 최상부상에 구축된 대화식 브라우저에 적용 가능하다. 다른 경우에, 프로토콜 식별자는 또한 다운로드하는 URL(플러그-인 또는 애플릿)을 가리키어 통신을 초기화한다. 이것은 또한 대화식 브라우저를 지원하는 클라이언트 플랫폼상에서 실행 가능한 로컬을 직접적으로 가리킬 수 있다. 예시적인 예는 분배 인식을 필요로 하는 선택이다(예를 들어, 인식은 원격 네트워크 상에서 수행됨)(IBM Docket No. YO999-113P에 서술된 바와같음). 다른 예는 다이얼로그를 수행하는 절차적인 기능(또는 로컬 기능)을 다운로드하는 선택을 포함한다. 또다시, 이들 대화식 절차는 CVM의 최상부상에 구축된다. 이들 개념은 단지 "선택"에만 유효한 것이 아니라 모든 "목표"에 대해 유효하다는 것을 이해하여야 한다.

< choice> 구성 요소의 "프롬프트 텍스트"는 포함하는 < menu> 가 방문되는 경우 말하여진다. < choice> 구성 요소의 프롬프트 텍스트를 정합하는 문법은 포함하는 < menu> 가 로딩될 때 활성화된 후 활성화 상태로 유지한다. 사용자가 < choice> 구성 요소의 프롬프트 텍스트에 정합하는 단어 또는 페이지를 말할때, 특정화된 목표가 방문된다. 예를 들어, 사용자는 이 선택으로부터 하나 이상의 유효한 단어보다 앞서 관심있는 단어 또는 페이지("컴퓨터", "가라" 또는 "선택")을 말함으로써 이들 선택들중 하나의 선택을 택할 수 있다. 이하에 메뉴 구성 요소가 예시될 것이다.

< menu>

주 메뉴로부터 선택하여 주십시오.

< choice target = "file:e-mail"> E-mail.< /choice>

< choice target = "file:news"> News.< /choice>

< choice target = "file:nav"> Navigation.< /choice>

< choice target = "file:mcform"> Food Ordering.< /choice>

< choice target = "file:weather"> Weather information.< /choice>

< choice target = "file:tutorial"> Tutorial.< /choice>

< /menu>

주 메뉴는 사용자가 대화식 브라우저로 회의를 시작할때 우선 들려지는 최상의 레벨 메뉴로서 서비스할 수 있다. 특히, 대화식 브라우저가 이 메뉴를 방문할때, 이것은 선택 목록 "E-mail, News, Navigation, Food Ordering, Weather information, Tutorial."에 앞서 "주 메뉴로부터 선택하여 주십시오"라는 말을 출력시킨다. 대화식 브라우저가 이 메뉴를 로딩하면, 사용자는 명령을 말함으로써 임의의 선택을 활성화(선택)시킬 수 있다(나머지 회의 동안). 허용 가능한 명령은 사용되는 입력 기술에 좌우된다. "당신이 듣고 있는 것은 당신이 말할 수 있다"라는 방식을 수행하는 일 실시예에서, 허용 가능한 명령은 프롬프트 텍스트, 예를 들어 "전자 사서함으로 가라" 및 "뉴스를 선택하라"의 일련의 단어보다 앞서 관심 페이지("가라" 또는 "선택과 같은)를 포함할 수 있다.

사용자가 선택을 말한 후, 대화식 브라우저는 소정의 선택(예를 들어, 목표 URL의 내용을 불러와서 CML 엔티티로서 해석되고 이 엔티티는 방문된다)과 관계되는 목표 속성에 의해 규정되는 목표 어드레스(예를 들어, URL, 소켓 어드레스)를 방문할 것이다. 상기 예의 메뉴에서, 모든 목표 URL은 이 경우에 파일인 포함하는 페이지의 URL, 데모/주 파일과 관계하는 URL에 상대적이라는 점에 유의하라. 예를 들어, 목표 "파일 : 뉴스" 는 주 메뉴를 포함하는 파일과 동일한 디렉토리에서 " 뉴스" 라 불리는 파일로서 해석된다.

요약하면, 사용자는 < menu> 구성 요소를 방문하기 위하여 < menu> 구성 요소의 타이틀 텍스트에 정합하는 페이지를 말할 수 있다. 사용자는 이 구성 요소에 의해 특정화된 목표를 방문하기 위하여 < choice> 구성 요소의 텍스트에 정합하는 페이지를 말할 수 있다. 사용자의 페이지는 텍스트에 나타나는(반드시 인접할 필요는 없다) 동일한 시퀀스의 텍스트로부터 하나 이상의 단어보다 앞서 관심 페이지를 포함하는 경우 < menu> 또는 < choice> 의 텍스트에 정합한다. 관심 페이지는 명령을 따르도록 지시하는 " 가라" 와 같은 페이지이다.

본문 구성 요소: < body> 구성 요소는 엔티티가 방문될 때 구어 출력으로 변환될 어떤 텍스트를 특정화하는 것이다.

< body name = " name" next = " target" >text< /body>

< body> 구성 요소가 방문될 때, 이것의 텍스트는 말하여지고 나서 다음 파라미터에 의해 특정화된 목표가 방문된다. JSML 마크업을 갖는 CML< body> 엔티티는 표시 지향된 마크업(가령 heading, 목록 등)을 갖는 HTML 본문 엔티티가 GUI를 위하여 제공되도록 하는 기능과 유사하게 음성을 위한 기능을 제공한다. 정보 검색 어플리케이션(가령 전자 사서함 및 뉴스)에 대해서, < body> 구성 요소는 통상적으로 사용자가 탐색하는 정보를 포함한다. 본문 구성 요소는 Java Synthesis Markup Language(JSML)을 사용하여 마크업되어 정확하고 효율적인 텍스트 대 음성 합성을 획득하는 것이 바람직하다.

< body>Text to be spoken, marked up using<EMP>JSML< /EMP> < /body>

< body> 구성 요소의 " 다음" 속성은 본문이 사용자에게 말하여진 후 방문되는 또다른 CML 구성 요소(가령 < menu> 구성 요소 또는 < form> 구성 요소)를 특정화하는 목표 어드레스(예를 들어 URL)인 값을 갖는다. 다음 예는 < body> 구성 요소의 기능 및 동작과 JSML 마크업 및 " 다음" 속성의 사용을 예시한 것이다.

< body next = " #menu1" >

< JSML>

음성 < EMP> 음성 브라우저< /EMP> 를 통해서 IBM에 오신걸 환영합니다. 이 지도는 여러분의 음성 브라우저를 제어하도록 사용할 수 있는 구어 명령으로 인해 여러분과 친숙할 것이다.

< /JSML>

< body>

이 본문은 JSML을 사용하여 포매팅되어 용어 " 음성 브라우저" 에 대한 엠파시스를 배치한다. 이 경우의 " 다음" 속성은 브라우저가 현재 페이지상의 명칭" menu1" 을 갖는 구성 요소를 방문하도록 지시하는 상대적인 URL(또는 목표 어드레스)이다.

전술한 바와 같이, CML의 일 실시예에서, < body> 및 < menu> 구성 요소는 구어 메뉴를 사용하여 정적 정보(static information)의 네비게이션 및 구어의 표시를 제공한다. HTML과 같이, 이 기능성 레벨은 인터링크된 CML 페이지 세트로서 조직될 수 있는 정적 정보에 대하여 충분하게 된다. 그러나, 사용자가 탐색 워드, 달러량, 데이터, 주식 명칭 등과 같은 큰 세트로부터 택한 정보를 어플리케이션에 제공할 수 있는 경우 훨씬 풍부한 어플리케이션 세트가 가능하다. 이와 같은 가능한 입력 세트는 너무 커서 메뉴에 제공될 수 없어, 클라이언트에서 입력을 수집하는 또다른 메카니즘이 필요로된다. 또한, 메뉴 및 본문 구성 요소와 같이 기억되기 보다 오히려 표시 서버에서 어플리케이션 응답 "on-the-fly" 를 계산하기 위하여 메카니즘이 사용될 수 있다. 이것은 예를 들어 서버상에서 실행되는 서브렛(servlet) 프로그램 및 CGI(Common Gate Interface)를 통해서 행해질 수 있다. 복잡한 논리가 트랜잭션 또는 서버 어플리케이션 이면에 감춰진 경우에, 이와 같은 서버측에서 응답 on-the-fly 수정이 진행되는 유일한 방법이다(논리에 관한 정보가 IBM Docket No. YO999-114에 전술한 대화식 스킴톤 방법을 사용하여 전송되는 경우 또는 정보가 클라이언트:구조 스킴톤에 전송되어 다이얼로그의 파트를 명확하게 하는 경우를 제외함).

이와 같은 사용자 입력을 수집하는 것은 또한 < form> 구성 요소를 사용하여 CML에서 성취될 수 있다.

형식 구성 요소: < form> 구성 요소는 사용자로부터 하나 이상의 정보 조각 또는 필드를 수집한다. < form> 구성 요소는 통상적으로 명칭, 어드레스, 전화 번호 및 메뉴의 선택 목록으로서 제공하는데 비실용적인 어떤 다른 타입의 정보와 같은 정보를 수집하는데 사용되고 다음과 같은 일반적인 형태를 취한다.

```
< form action = " URL " >
```

서두 메시지 텍스트

```
< field name = "name1" rule ="JSGF">prompt text1< /field>
```

```
< field name = "name2" rule ="JSGF" >prompt text2< /field>
```

...

```
< /form>
```

동작은 < form> 에서 사용되는 목표 URL이다. 이 형식의 < field> 값은 HTML과 동일한 방식으로 속성 값 쌍으로서 이 동작에 부가된다. 형식은 사용자 입력 세트 모두를 그룹화하는데, 이 세트 모두는 표시 서버(차례로 요청을 정보 서버에 전송할 수 있다)로 다시 진행시키는 것을 보장하여 어플리케이션 응답을 얻도록 하는데 충분하다. &lt; form> 구성 요소가 방문될 때, 이것의 타이틀 텍스트(서두 텍스트)가 말하여지고 나서 사용자는 임의의 포함된 < field> 구성 요소의 값에 대하여 하나씩 프롬프트된다. < form> 구성 요소의 타이틀 텍스트에 정합하는 문법은 이 < form> 이 로딩될 때 활성화된 후 활성화 상태로 유지시킨다. < field> 태그는 다음과 같은 일반적인 형태를 취한다.

```
< field name = " name" rule = " ruleName" value = " value" >prompt text< /field>
```

값이 (사용자 또는 < field> 태그의 값 파라미터중 어느 하나에 의해) 이미 제공되어 있지 않고 프롬프트 텍스트가 비워있지 않는 경우, 사용자는 특정화된 프롬프트 텍스트를 말함으로써 필드의 값에 대하여 프롬프트된다. 필드를 채우는 수용가능한 응답은 사용되는 입력 기술에 좌우될 것이다. " 당신이 듣는 것은 당신이 말할 수 있다" 라는 방식에 대하여, 수용 가능한 응답은 특정화된 문법(이것은 필드에 대한 가능한 값의 세트를 제공하는 Java Speech Grammar Format(JSFG)에서 바람직하게 된다.)에 정합하는 페이지보다 앞서 프롬프트 텍스트의 일련의 워드를 포함한다. 후술

하는 바와 같이, 이것은 NLU 입력으로 확장될 수 있다. 사용자의 응답이 필드를 채우는 경우, 사용자는 다음의 채워지지 않은 필드 등에 대해 프롬프트된다. 모든 필드 값이 특정화된 경우, 동작 목표는 예를 들어 HTTP GET 방법(즉, 결과가 표시 서버에 전송됨)을 사용하여 방문되며, CML의 페이지는 엔티티를 포함하여 복귀되고 나서 방문된다. 사용자는 프롬프트될 때 또는 필드의 합법적인 값(legal value)보다 앞서 프롬프트의 일부를 말함으로써(즉, 룰 파라미터에 의해 특정화된 룰 명칭에 정합하는 페이지보다 앞서 있는 프롬프트 텍스트를 말함으로써, 여기서 "룰 명칭"은 완전히 자격이 있는 JSGF 룰 명칭이다) 형식이 여전히 범위내에 있는 어떤 시점에서 임의의 필드값으로 채울수 있다.

다음 예는 형식 구성 요소의 기능 및 동작을 예시한 것이다.

```
< form action = " HTTP://localhost:8080/servlet/McServlet/" >
```

주문 형식을 완성하여 주십시오

```
< field name = " sandwich" rule = " mcgrammar.sandwich" >Sandwich is ?< /field>
```

```
< field name = " drink" rule = " mcgrammar.drink" >Drink is ?< /field>
```

```
< form>
```

상기 형식이 방문될 때, 대화식 브라우저는 서두를 출력한다. " 주문 형식을 완성하여 주십시오 ". 그리고 나서, 사용자는 " 샌드위치가 있는가?" 라고 프롬프트된다. 그리고 나서, 사용자는 이하를 토대로 " 샌드위치는 치킨 샌드위치" 또는 " 샌드위치는 햄버거" 라고 응답한다. 특정화된 JSGF 룰 " <mcgrammar.sandwich>가 JSGF 문법 " mcgrammar" 에 서 찾아진다.

이 JSGF 문법 " mcgrammar" 는,

문법 mcgrammar;

대중적인< 샌드위치> = 햄버거 치킨 샌드위치;

대중적인< 음료> = 코크 펄시; 판독한다.

사용자는 또한 이것이나 이 값에 대하여 프롬프트되기 전 그리고 샌드위치 형식이 활성 상태를 유지하는 동안 " 샌드위치는 샌드위치이다" 또는 " 음료는 음료이다" 라고 말함으로써 임의의 필드를 채울수 있다. 이 경우에, 브라우저는 필드에 대한 값을 선택하고 나서 채워지지 않은 필드에 대하여 하나씩 사용자에게 프롬프트한다. 이것은 " 혼합된 주도권(mixed initiative)" 으로서 공지된 다이얼로그 특징이다. 이것은 사용자가 " 샌드위치는 샌드위치이다" 라고 말함으로써 주도권을 가질 수 있거나, 시스템이 예를 들어 사용자가 음식 주문 형태를 이끄는 메뉴 선택을 활성화시키기 때문에 샌드위치에 대한 사용자를 프롬프팅함으로써 주도권을 가질 수 있다는 것을 의미한다.

CML 형식은 HTML 형식과 유사하고 일 실시예에서, CML은 HTML과 유사한 메카니즘을 사용하여 필드값을 서버에 전송한다: 사용자가 각각의 필드에 대한 값을 특정화하는 이 형식을 완성할때, 브라우저는 각각의 필드에 대한 명칭=값 쌍을 특정화된 형식 동작 URL에 부가하고 최종적인 URL은 HTTP를 통해서 서버로부터 요청된다. 형식 동작 URL은 CGI 스크립트 또는 자바 서브렛과 같은 어플리케이션 - 특정 기능의 명칭을 포함하는 것이 바람직한데, 이 서버는 명칭=값 쌍을 처리하여 응답을 복귀하도록 호출할 것이다.

완성된 형식에 대한 응답이 사용자에게 디스플레이로 제공될 HTTP의 페이지가 되는 그래픽 웹 브라우저의 경우와 같이, 대화식 브라우저에서, 이 응답은 음성(특정화된 본문, 메뉴 또는 형식 엔티티를 방문함으로써) 사용하여 사용자에게 제공될 CML의 페이지가 될 것이다. 각각의 경우에, 새로운 페이지는 다음의 입력 동작의 해석에 영향을 미칠 것이다.

다이얼로그 이슈: 구어의 다이얼로그 어플리케이션의 설계시 고려할 한가지 사항은 사용자가 "대화"를 어떤 시점에서 무엇을 말할 수 있는지를 손쉽게 판단하도록 하는 것이다. 이상적으로, 어플리케이션은 사용자(NLU)로부터 어떤 이유 있는 구어 입력을 수용하는데, 이것은 때때로 수행하기 곤란하였다. 대신에, 어플리케이션은 제한된 입력 세트를 수용하도록 설계될 수 있다. 그러나, 이 설계는 사용자 자신의 원래 구어의 임의의 서브셋을 포함하는 "언어"를 학습 및 기억하여야 하는 가중한 임무를 사용자에게 부과한다.

CML 브라우저는 다양한 방식을 수행함으로써 이 문제를 완화시키는데 유용하다. 한가지 방식은 "당신이 듣는 것은 당신이 말할 수 있다"라는 방식과 관계되는데, 이 방식에서(전술한 바와 같음) 수용 가능한 구어 입력은 항상 사용자에게 제공되는 구어 프롬프트를 에코시킨다. CML 구성 요소가 방문될 때마다, 브라우저가 수용될 사용자 입력 세트는 변경된다. 예를 들어, 사용자가 < menu> 또는 < form> 을 방문한 후, 브라우저는 메뉴 또는 형식에 적절한 응답을 수용할 것이다. "당신이 듣는 것은 당신이 말할 수 있다"라는 방식 다음에, 수용된 입력은 일반적으로 메뉴 또는 형식 프롬프트의 에코되거나 어떤 경감된 프롬프트의 버전이 된다.

예를 들어, 사용자가 "주 메뉴로부터 선택하라: 전자 사서함, 주식 지수"를 듣는 경우, 사용자는: "주 메뉴로 가라", "주식 지수로 가라" 및/또는 "전자 사서함으로 가라"라고 말할 수 있다. 또한, 사용자가 "당신은 2개의 메시지, 즉 골프 티 타임에 관한 빌; 스미스로부터의 새로운 메시지 1, 프로젝트 회의에 관한 제인 존스로부터 메시지 2를 갖고 있다"라고 듣는 경우, 사용자는 "새로운 메시지로 가라", "새로운 메시지 1로 가라", "메시지 2로 가라", "빌로부터의 메시지로 가라", "티 타임에 관한 메시지로 가라" 및/또는 "프로젝트에 관한 메시지로 가라"라고 말할 수 있다. 또한, 사용자가 "주식 지수. 주식 심벌이 있는가?"라고 듣는 경우, 사용자는 "주식 지수로 가라", "지수로 가라", "주식 심벌은 IBM이다", "심벌은 IBM이다" 및/또는 "주식은 IBM이다"라고 말할 수 있다.

전술한 바와 같이, 상기 2개의 제1 실시예는 몇몇 표제어("주 메뉴로부터 선택")와, 몇 개의 프롬프트 문서(prompt text)("e-메일(E-mail)", "주식 시세(stock quote)")를 포함하는 복수의 < choice> 요소를 이용하는 CML로 언어될 수도 있다. 최종 실시예는 몇몇 표제어("주식 시세(stock quotations)")와 몇몇 프롬프트 문서("주식 부호인가")를 포함하는 소정 < field> 요소를 포함하는 < form> 요소를 이용하여 언어될 수도 있다. 따라서, < menu> 또는 < form> 의 표제어와 < choice> 또는 < field> 의 프롬프트 문서는 사용자가 말할 것을 정의한다.

다이얼로그를 실행하기 위한 기타 옵션이 사용될 수도 있다. 예를 들면, 몇 가지 양식에 대한 응답을 미리 로딩할 수 있다. 즉, 소정 요소의 스크립팅이 절차를 통한 사전 프로그래밍, 응답 기반 스크립트, 사용자로부터의 선택(즉, 변수는, 예를 들어, 프롬프트의 필드)에 도입될 수 있다. 물론, 이 스크립팅은 기타 이벤트 및 절차 로컬에 의해서 할당된 변수, 또는 배포된 변수(예를 들어, 표시 서버 등으로의 문의에서 비롯된, 날짜)를 사용하거나 참조할 수 있다. 이것은 이벤트 포착, 논리적 동작의 부가, 루프 및 방향 지정문(redirection statement)(예를 들어, go to)을 실행한다. 복수의 양식을 사용하는 상태에 대해서, 이벤트 포착 능력이 요구된다. 그러한 CVM 문맥 내의 이벤트는 애플릿 및 기타 대화식 절차, 또는 스크립트로부터 야기된 이벤트도 포착할 수 있다. 스크립팅은 자바스크립트(Javascript)나 이시엠에이스크립트(ECMAScript)와 유사한 방식으로 수행될 수 있다. 스크립트는 < script> ...< /script>의 양식 내에 삽입될 수 있다. 또한, 이것은 양식, 메뉴 등이 이벤트를 의심하거나 포착할 수 있음을 의미한다. 이것은 적절한 프로그래밍 능력을 부여한다. 스크립트는 CML 페이지 내의 어디에도 삽입될 수 있다. 절차, 다이얼로그 구성 요소, 대화식 기

반 계층 및 기타 서비스는 < object> ...</object> 태그에 의해서 실행될 수 있다.

다른 실시예에 있어서, " 청취한 것을 말하십시오" 라는 방안은 (전술한 바와 같이 브라우저에 의해서가 아니라) 사용자가 대화를 제어할 수 있게 한다. 보다 구체적으로는, 대화식 세션 중에, 청취되었던 모든 이전에 방문된 < menu&gt; 나 < form> 은 활성화된 상태로 유지된다. 실시예로서, 주메뉴를 들은 사용자가 " e- 메일" 및 " 뉴스" 에 대해서 선택한다고 가정한다. " 청취한 것을 말하십시오" 라는 방안으로, 사용자는 항상 주메뉴 청취에 수반되는, " 전자 우편 선택" 이라고 말함으로써 즉시 e- 메일 어플리케이션을 선택하거나, 주메뉴로 복귀할 필요없이 " 뉴스로 이동" 이라고 말함으로써 뉴스 어플리케이션을 선택할 수 있다. 사용자가 주메뉴상의 선택을 잊어버린 경우, 사용자는 " 주메뉴로 이동" 이라고 말함으로써 주메뉴로 돌아갈 수 있다. 이 원리는 사용자가 들은 모든 메뉴 및 양식에 적용된다.

다른 원리는 " 청취할 것을 말하십시오" 이다. 여기에서 메뉴 및 양식은 방문되기 이전에 활성화되어 브라우저 세션 내에 활성화 상태로 유지된다. 보다 구체적으로, 그것이 브라우저에 의해서 로딩될 때 몇 가지 메뉴와 양식이 활성화되며, 이것은 몇가지 메뉴와 양식을 포함하는 페이지가 로딩되는 경우와 같이 메뉴와 양식이 방문되기 이전에 일어날 수도 있다. 그 페이지 상의 메뉴나 양식중 하나만 방문되더라도(URL의 특성에 따라서), 일반적으로 그 페이지의 모든 메뉴나 양식이 활성화된다.

이것은 다음 실시예에서 설명된다.

< page>

< body text=" #new2menu" >

New message 2 form Steve about Pay raise.

Bruce,

We ve decided to give you a 250% pay increase this year.

Don t spend it all in one place.

Steve

< /body>

< menu name=" new2menu" >

Choose from the message menu

< Choice target=" #new3" > Next message.< /choice>

< Choice target=" #new2forward" > Forward message.< /choice>

< /menu>

< /page>

상기 실시예에서, 사용자는 메시지 2를 듣기 원한다고 가정한다. 대화식 브라우저는 이 경우에 있어서 메시지의 본문인 페이지에 대한 URL을 방문할 것이다. 즉, 페이지를 인출하여 그 페이지 상의 제1 유닛을 방문할 것이다. 이 메시지가 완료된 후, < body> 요소의 " 다음" 속성은 대화식 브라우저가 " new2menu" 를 방문하도록 하여, 사용자에게 메시지의 배치에 대한 한쌍의 선택을 제공한다. 그러나, 사용자는 기다릴 필요 없이, 선택 요소 중 하나를 고르기 전에 메뉴를 듣는다. 대신에, 페이지가 먼저 로딩되면 메뉴가 활성화되기 때문에, 사용자는 말하기 위한 소정 시간, 예를 들어 " 다음 메시지로 이동" 이라고 말하는 소정 시간에 메시지를 중단할 수도 있다. 이 특징은 사용자가 아는 자주 사용하는 어플리케이션에서 아직 표시되지 않은 메뉴 및 양식을 예측하는 데 유용하다. " 청취할 것을 말하십시오" 라는 방안은, 대화식 브라우저가 대화식 기반 계층을 이용하여 상기 CVM을 구축할 경우 다이얼로그 객체를 미리 로딩하기 위해서, 위에서 편집된 IBM Docket No. Y099 - 114에 설명된 바와 같은 기술을 이용하여 구현되는 것이 바람직하다. 그 대신에, 대화 객체나 대화 구성 요소는 CML로 구현할 수 있다.

" 청취할 것을 말하십시오" 및 " 청취할 것을 말하십시오" 라는 메카니즘은, " 혼합 개시(mixed initiative)" 라고 알려진 다이얼로그 특징에 적용되는 것이 바람직하다. 이것은 몇 가지 경우에 컴퓨터가 대화 중 개시하고 - 예를 들어 브라우저가 메뉴 항목 세트에 대해 사용자에게 조언할 경우- 와, 가끔 사용자가 개시- 예를 들어, 사용자가 프롬프트를 무시하고 과거에 들었던 메뉴 항목을 선택하거나, 컴퓨터를 일시 중단시키고 아직 표시되지 않은 메뉴 항목을 선택할 경우- 하는 것을 말한다. 따라서, 브라우저에 의해서 수용할 수 있는 음성 응답 세트는, 가장 최근에 방문된 요소에 의해서만이 아니라, 브라우저에 의해서 방문 및/또는 로딩된 전체 CML 요소 세트에 의해서 정의된다. 그러나, 이것은 가끔 더 이상 관련이 없는 활성화된 메뉴나 양식으로 인해서 예기치 못한 결과를 초래할 수 있다. 따라서, 본 발명에 따른 CML은, 메뉴 및 양식에 대한 활성화 범위를 제한하는 메카니즘을 제공한다.

범위(Scope): 범위는 CML 어플리케이션 설계자가 특정 < menu> 또는 < form> 에 관련되어 말해진 입력이 활성화되는 기간을 제어할 수 있도록 하는 특징이다. 이 " 범위" 특징은 다음과 같은 < menu> 또는 < form> 요소와 연관될 범위 속성에 의해서 지정된다.

```
< menu name=" name" scope=" scope" > title text, choices < /menu>
```

```
< form name=" name" action=" target" scope=" scope" > title text, fields< /form>
```

" 범위 속성의 바람직한 값은 다음과 같다.

로컬(Local): 메뉴나 양식 자신이 브라우저가 방문된 최종 요소인 경우에만 메뉴나 양식이 활성화된다.

페이지(Page): 브라우저에 의해서 방문된 최종 요소가, 메뉴나 양식을 직접 포함하는 페이지(또는 그 페이지의 서브 페이지) 상에 있는 경우에만 메뉴나 양식이 활성화된다.

글로벌(Global): 이것은 디폴트 값이며, 여기에서 메뉴나 양식은 브라우저에 의해서 먼저 로딩될 때 시작하는 브라우저 세션의 전체 기간 동안 활성화된다.

어플리케이션 특성(Application specific): 이것은 사이트나 어플리케이션(예를 들어, 메타 정보 태그(meta - information tag)를 통해서)으로 식별되는 모든 페이지를 참조한다.

여러 경우에 있어서, 먼저 브라우저가 메뉴나 양식을 접한 직후에 메뉴나 양식이 활성화되어 이후 활성화된 상태로 유지될 경우, 사용자에게 최고의 융통성이 제공된다. 예를 들어, 전술한 바와 같이, 먼저 복잡한 메뉴의 경로를 통해서 복귀할 필요없이, 메뉴에 의해서 제공되는 선택 요소로 사용자가 직접 점프할 수 있도록, 전체 세션 동안 주 메뉴나 주요 주제 메뉴가 활성화되는 것이 바람직하다. 그러나, 몇몇 상황에 있어서 이것은 음성 인식 정밀도를 감소시킬 수 있고 예

기치 않은 결과를 초래할 수 있다.

다음 예제를 고찰해 보자.

```
< page name=" new2" >
```

```
< body next=" #new2menu" >
```

New message 2 from Steve about Pay raise.

Bruce,

We ve decided to give you a 250% pay increase this year.

Don t spend it all in one place.

Steve

```
< /body>
```

```
< menu name=" new2menu" scope=" page" >
```

Choose form the message menu.

```
< choice trget=" #new3> Next message.< /choice>
```

```
< choice target=" #new2forward" > Forward message.< /choice>
```

```
< /menu>
```

각 메시지에 관련된 메뉴는 사용자가 " 다음 메시지로 이동" 이라고 말할 수 있게 한다. 그러나, 이 선택은 사용자가 e - 메일 메시지를 벗어나서 다른 업무를 수행하는 것을 감지하기 어렵게 만들 것이고, 만일 음성 인식 오류로 인하여 활성화되었을 경우 매우 놀랄수도 있다. 따라서, 본 실시예에 있어서, 메시지 메뉴는 " 페이지" 의 범위를 부여받는다. 이것은, 단지 브라우저에 의해서 방문된 최종 요소가 메시지 메뉴를 포함하는 페이지 내에 있는 한 그것이 활성화될 것이라는 말이다. 이 경우에, " 다음 메시지로 이동" 이라는 말은 단지 최종 브라우저 동작이 e - 메일 항목의 본문의 방문을 수반할 때까지만 활성화된다는 것을 의미한다.

범위의 부가적인 사용 및 범위에 대한 행동은, (후술하는 바와 같이) 개선된 CML로 제공된다. CML과 함께 사용될 다른 범위 태그는 다중적(multiple)이다. (CML 양식이 아닌) 다중 양식의 사용에 대한 논의는 Kishore, et al, " Free - Flow Dialog Management Using Forms," Proc. Eurospeech 1999, Budapest Hungary, September 1999 및 Davies et. al., " The IBM Conversational Telephony System For Financial Applications, Proc. Eurospeech 99, Budapest Hungary, September 1999에 개시되어 있다. 이에 따라서, 활성화된 다중 페이지를 구비하는 대신에, 복수의 양식이 한 페이지 상에서 동시에 활성화될 수 있다는 점에 유의해야 한다. 이들 페이지는 문서상에 명백히 겹쳐질 수 있고 (로딩될) 주소에 의해서 참조될 수 있다. 유사하게, 이들 양식 중 몇몇은 절차상의 구성 요소로서 표시될 수 있다.



전술한 바와 같이 전형적으로 소정 NL 양식하에서 NL 다이얼로그는 다중 양식의 활성화에 의해서 실행된다. 그 양식은 소정 양식의 모든 필수 필드가 채워질 때까지 동시에 채워진다. 다중 양식이 채워지면, (예를 들어, 신규 양식의 활성화에 의해서) 명료한 대화가 부가된다. 전형적으로 다중 범위 양식은 다음과 같이 상이한 양식을 로딩하는 것을 나타내는 페이지를 통해서 로딩된다.

```
< menu name=mainnl scope=global>
```

Introduction dialog

```
< form name=form1 scope=multiple> ...< /form>
```

```
< form name=form2 scope=multiple> ...< /form>
```

```
< form name=form3 scope=multiple> ...< /form>
```

...

```
< /menu>
```

동작을 암시하는 소정 입력에 대해서, 그 동작이 완료된 후 새로운 페이지가 로딩될 것이다.

```
< menu name=mainnl scope=global> (we " update" the global menu)
```

```
< form name=form1 scope=multiple context=maintain> ...< /form>
```

```
< form name=form2 scope=multiple context=reset> ...< /form>
```

```
< form name=form3 scope=deactivated context=reset> ...< /form>
```

```
< form name=formnew scope=multiple context=reset> ...< /form>
```

...

```
< /menu>
```

대신에, 모든 이들 양식은 다음과 같이 로딩될 수 있다.

```
< form name...scope...load=target> < /form>
```

로드 인자가 존재하면, 양식의 내용이 대상으로부터 다운로드되어 < form> < /form> 태그 사이에 배치된다. 임의의 어떤 다른 요소에 대해서도 동일한 개념이 사용될 수 있다. 전형적으로, 메뉴는 갱신되어 신규 양식이 부가되며, 몇몇 양식은 해제된다(범위 해제).

전술한 예제내의 " 문맥(context)" 태그는 과거 대화 이력(history)을 처리하는 방법을 나타낸다. 위에서 주어진 두 실시예에서, 문맥 " 유지" 를 포함하는 양식은 대화의 이전 단계(이전에 사용자에게 의해서 입력된)에서 이 양식내에 채워진 값을 유지할 것이다. " 리셋" 태그는, 값을 디폴트값으로 복원하는 것을 나타낸다. 또한 범위는 scope = 5s or scope = 1 minute or scope = the name of a scope와 같이 기간 스탬프로 지정될 수 있다는 데 유의해야 한다.

범위는 다음과 같이 도입된다.

< scope name = scope1 status = active> < /scope>

정의되지 않을 경우 범위는 비활성화된다. 정의될 경우, 범위는 활성화되거나 비활성화된다.

명칭으로 범위가 지정되는 요소는 범위가 활성화되거나 비활성화될 때 활성화된다.

본 명세서에서 사용되는 양식은 다음과 같은 절차상의 객체로 이용될 수 있음을 알 수 있을 것이다.

< form name=form3 scope=multiple context=reset load=target> < /form>

여기에서 대상 페이지는 다음과 같이 애플릿, 절차, 서비스 등을 포함한다.

< Dialog Object>

< applet...>

< /applet>

< /Dialog object>

또, 다이얼로그 객체는 애플릿(자바), (브라우저에 의해서 해석된) 스크립트, 대화식 객체(대화식 기반 계층을 이용하는 CVM의 최상위에 기재된 대화식 다이얼로그), 플러그인이나 서브릿 활성화, 또는 기타 절차상의 구현이나 서비스 공급자가 될 수 있다.

이 경우에, 다중 객체는 병렬로 로딩될 수 있다. 전형적으로, 이것은, 브라우저가 CVM의 최상위에서 구현되는 경우, CVM이 이들 상이한 객체 및 그 문맥을 등록하여 다이얼로그가 사용자의 입력 기능으로서 활성화되는지를 판정할 수 있는 경우 등이다.

이 최종 항목은 NL 기술에 관계없이, 적절한 다이얼로그 관리 능력이 있는 CML 및 브라우저로 설명한다. 여기에서, CML은 NL 다이얼로그를 설계하는 데 사용될 수 있다. 양식 기반, 절차상의 기반(결정 네트워크) 및 문법 기반 NL 다이얼로그 관리자는, 이 방안에 의해서 충분히 지원된다.

또한, 오버로딩된 태그(즉, 태그와 관련된 동작의 변경)를 가질 수 있다. 전형적으로, 이것은 XML 또는 오버로딩 정의의 범위 및 태그의 새로운 의미를 설명할 수 있는 기타 메커니즘에 의해서 수행될 수 있다.

브라우저 특징: 지금까지 CML 및 마크업 언어(markup language)에 응답해서 브라우저가 어떻게 행동하는 지에 대해서 음성 입력 및 출력을 고려하여 설명하였다. 후술하는 설명은 CML의 본질이라기 보다는 브라우저 구현의 특징인 대화식 브라우저의 특징에 대한 것이다.

먼저, 대화식 브라우저는 (시각적인 브라우저의 메뉴 상에 포함되는 내장 기능 및 툴바와 유사한) 복수의 내장된 명령으로 구현되는 것이 바람직하다. 그러한 명령의 예는 다음과 같다.

대기([be] quiet), 폐쇄(shut up): 이들 명령은 브라우저가 현재 구두로 전달된 출력이나 사용자로부터의 부가 명령에 대한 대기를 정지시키도록 할 것이다. 이 특징은 "바지인(barge-in)"이라고 알려져 있고, 사용자가 언제라도 브라우저를 일시 정지시킬 수 있도록 모든 음성 입력에 제공한다. 그래서, 예를 들면, 사용자가 "대기"라고 말해서 브라우저를 일시 정지시킬 수 있고, 이 경우에 브라우저는 현재 출력 및 추가 명령에 대한 대기를 중단할 것이다. 브라우저가 일시 중단된 출력을 반복하도록 하기 위해서, 사용자는 예를 들어, "반복"이라고 말할 수도 있다. 또한, 사용자는 어느 지점에서든 메뉴 항목 선택 등의 음성 명령으로 브라우저를 일시 정지할 수도 있다.

재언급(say again), 반복(repeat that): 이들 명령은 브라우저가 가장 최근에 방문된 요소(메뉴, 양식 또는 본문)를 반복하도록 할 것이다.

복귀(go back): 이 명령은 브라우저가 이력 목록(history list) 내의 이전에 방문된 요소로 돌아가도록 할 것이다.

전진(go forward): 이 명령은 브라우저가 이력 목록 내의 그 다음 방문된 요소로 진행하도록 할 것이다(몇 번의 "복귀" 명령 후에만 적용됨).

선두로 이동(go to the beginning): 이들 명령은 브라우저가 방문된 최초 요소(예를 들어, 홈 페이지)로 이동하도록 할 것이다.

상기 예시적인 명령 내의 괄호는 취사 선택어(optional words)임을 의미한다. 이들 명령 중 몇몇은 "부탁합니다(please)" 라는 말이 앞이나 뒤에 붙을 수 있다. 기타 전형적인 명령은 브라우저용으로 정의되거나 사용자에게 의해서 부가될 수 있다.

필연적으로 이들 다양한 유형의 음성 입력 사이에 다의성이 발생할 것이다. 이 다의성 해결은 유닛(양식 및 메뉴)의 가장 최근에 사용된(MRU: most-recently-used) 큐우(queue)를 유지함으로서 유닛 레벨에서 처리한다. 보다 구체적으로, 유닛(다음에 표제어나 선택 프롬프트가 오는 유의 어구(attention phrase), 다음에 합법적인 필드값이 오는 필드 프롬프트)에 의해서 허용된 음성 어구와 부합될 경우, 음성 입력은 유닛에 대해서 분석된다. 브라우저는 각 유닛에 대비하여 모든 음성 입력을 순서대로 MRU 큐우 내에 분석하려고 한다. 소정 실시예에 있어서, MRU 큐우는 다음과 같이 보존될 수도 있다.

1. 큐우 내에서 음성 유닛이 소정 유닛에 대해 분석되면, 그 유닛을 큐우의 헤드로 이동한다.
2. 그 다음, 음성 입력이 방문될 대상을 발생하면, 그 대상이 포함된 페이지 내의 모든 유닛을 그 큐우의 헤드로 이동한다.
3. 최종적으로, 방문된 대상에 대응하는 유닛을 그 큐우의 헤드로 이동한다.

브라우저가 CVD의 최상위에 마련되기 때문에, IBM Docket No. YO999 - 111P 내에서 논의된 바와 같이 다의성의 발생이 의심될 경우, CVM이 적절한 다의성 해결 및 사용자와의 다이얼로그를 수행할 것이라는 점을 알 수 있다. 본질적으로, CML 규격이 생성된 적이 있으면, 그것은 다이얼로그가 처리될 방식을 실질적으로 지시할 것이다. 이것은 중복될 필요가 있거나, 기타 행동이 도입될 필요가 있을 경우, (가능할 경우) 기반을 이루는 CVM, OS 또는 플랫폼으로서 빠른 호출에 의해서 수행될 수 있다.

플러그인(Plugins): 본 발명은 플러그인 또는 애플릿류의 능력을 제공하는 메커니즘을 제공한다. <choice> 대상이나 <form> 동작을 지정하는 URL은, 다음의 몇가지 예시적인 양식 중 하나가 이어지는 로컬 프로토콜을 사용하여 플러그인 코드를 호출할 수도 있다.

local : service/function, 또는

local : service/function?arg1=value1,arg2=value2...

첫 번째 양식은 인자가 없는 <choice> 대상 및 브라우저에 의해서 포함된 <field> 들의 값으로부터 인자가 제공되는 <form> 동작에 대해서 유용하다. 두 번째 양식은 인자가 URL로 제공되는 <choice> 대상에 대해서 유용하다.

자바를 이용하여 구현되는 브라우저의 바람직한 실시예에 있어서, 로컬 : URL은, 부류명으로 매핑하는 로컬 테이블내의 서비스를 검색하고, 미리 실증된 적이 없는 경우 그 부류의 대상을 실증한 후, 호출될 방법명으로서 함수를 해석함으

로써 구현된다. 서비스 테이블은 대화식 브라우저의 파일 서비스 내에 배치되는 것이 바람직하며, 파일 서비스 자신은 자신의 형식에 대한 설명을 포함한다. 이것은 모든 가능한 서비스 및 (절차상의 대상이든 아니든 간에) 기타 대상의 등록을 포함한다.

플러그인 코드는 로컬에 의해서 호출된다 : URL은 URL로 표현되는 CML 문서를 포함하는 문자열로 복원된다. 로컬 방문 : URL은 복원된 문서가 소정의 다른 URL의 방문과 동일한 방식으로 해석되도록 한다. 또한, 이들 특징은, 로컬 어플리케이션으로서 동작하도록, (다운로드되거나 로컬의) 애플릿, 대화식 객체 및 서브렛/시지아이(servlets/cgi) 뿐만 아니라 (예를 들어, 소켓 커넥션 및 RPC 프로토콜을 갖는) 배포판 어플리케이션을 이용하여 구현될 수도 있는 것이 바람직하다.

이 서비스는, 스레드(thread)를 개시하고, 브라우저에 의해서 제공된 API를 통해서 (로컬: URL을 포함하는) 지정된 URL을 브라우저가 비동기적으로 방문하도록 함으로써, (예를 들어, 비동기적으로 중요한 이벤트를 사용자에게 통보하도록) 브라우저와 비동기적으로 상호 작용할 것이다. 사용자가 브라우저가 URL을 방문하도록 하는 무언가를 말한 경우와 마찬가지로, 브라우저 API는 로컬 플러그인 코드를 허용하여, 브라우저가 지정된 URL을 방문하도록 한다. 이것은 로컬 이벤트에 근거한 비동기적 사용자 통보를 제공하는 데 사용될 수 있다. 호출되었을 때, 브라우저가 다음과 같이 지정된 매개변수로, 지정된 URL을 방문하도록 하는 방문 방법을 로컬 플러그인이 계승함으로 인하여, 로컬 플러그인은 CML류, 서비스로 확장된다: 공용 등급 서비스[public class Service{public void visit(URL url, Dictionary parameters) throws IOException;}]

삽입된 대화식 브라우저 어플리케이션에 대한 고찰

등록 메카니즘(Registration mechanism): 삽입된 대화식 브라우저의 기능이 부적절한 자원(메모리, CPU 전원 등)으로 인해 제한되는 경우에, 브라우저가 그 능력에 대한 설명을 서버에 전송하도록 하는 쿠키(cookie) 메카니즘이 사용될 수 있다. 예를 들면, Speech\_Reco=1;Speaker\_Reco=0; TTS=1; Dialog\_Manager=0, NLU=0; Speech\_Reco.Vocabulary\_Size=500; Speech\_Reco.FSG=1 등이 있다. 상기 쿠키를 수신하면, 서버측은 그 내용을 검토한 후 CML 페이지를 적절히 수정한다. 예를 들면, 만약 쿠키가, 상기 어휘 사이즈가 50임을 나타내면, CML 메뉴는 아주 제한된 어휘(즉, 실제 링크 대신에 항목당 번호의 선택)를 갖도록 생성될 수 있다.

다른 실시예에서는, 쿠키를 이용하지 않고, 전술한 특허 출원 IBM Docket Nos YO999 - 111P 및 YO999 - 113P에 기술되어 있는 등록 대화식 프로토콜이 이용될 수 있다. 상기 실시예에서, 상기 엔진의 특징 및 특성과, 가능하게는 어플리케이션의 요건을 설명하는 객체들이 교환된다. 핸드셰이크는 상기 어플리케이션들의 대화식 책임을 결정할 것이다. 또한, 그러한 경우, 상기 핸드셰이크는 HTTP 또는 RPC, RMI 등과 같은 기타 프로토콜을 이용할 수 있다. 이것은 특히 서버와 직접 대화하여 그들의 동작이 로컬 브라우저의 능력에 맞는지를 검사할 수 있는 다이얼로그 객체에 있어 중요하다. 종래의 브라우저들에서와 같이, 이들 쿠키(또는 절차 중심(procedural) 프로토콜)은 표시의 주문 또는 사용자 기호에 맞춘 서비스 또는 이들 기호 또는 세션 이력의 유지와 같은 다른 작업들을 수행할 수 있다. 즉, 만약 그것이 사용자에게 의해 선택된 옵션이라면, 문맥 이력의 장치 유지는 접속시 업로드된 사이트의 과거 이력을 갖는 쿠키를 가짐으로써 보장된다. CVM 또는 절차 중심 객체들을 통해 수행될 때, 프로토콜을 통해 정보가 전송된다. 한편, 서버에 저장될 때 장치 유지를 위해 IC 쿠키가 또한 사용된다. 상이한 장치들로부터 접속이 이루어지면, 상기 쿠키(절차 중심 프로토콜)는 서버 또는 표시 서버에 저장될 수 있다. 사용자의 식별은 사용자 ID, 호출자 ID, 화자 인식 또는 음성 생체 측정(speech biometrics)과 같은 종래의 기술을 이용하여 행해질 수 있다. 다른 메타 정보(유지, 사용자 기호, 어플리케이션 기호, 및 이력, 문맥 등)가 전송되어야 한다(쿠키 또는 종래의 프로토콜을 통하여).

따라서, 만약 대화식 엔진이 기본 형태를 생성할 수 없다면, 태크가 기본형을 제공할 수 있다. 다음과 같은 여러 가지 방법이 이용될 수 있다. 1) 브라우저 능력의 결정시, 브라우저는 상이한 CML 페이지를 전송한다. 2) 트랜스코더는 등록 된 능력에 기초하여 브라우저에게 전송된 CML 페이지를 능동적으로 수정할 수 있다. 3) CML 페이지는 음성을 전송하고 네트워크로 연결된 서버(IBM Docket No. YO999-113P에서 논의)상에서 복잡한 기능을 수행하도록 수정될 수 있다.

동적 비활성화(Dynamic deactivation): 대화식 브라우저는 CML의 동적 활동을 이용할 수도 있다. 이것은 포함된 어플리케이션들에 대한 대화식 브라우저의 동작의 변경을 허용하는 다른 해결책이다. 특히, "당신이 듣는 것은 당신이 말할 수 있는 것이다"라는 패러다임 대신, 대화식 브라우저는 CML에 의해 명시적으로 제한되지 않으면 국부적으로만 마크업(markup)할 수도 있다. 또한, 지원된 활성 어휘(vocabulary)가 최대량에 도달하면, 음성 브라우저는 점진적으로 그리고 계층적으로 비활성화될 수 있다: 브라우저 섹 명령 또는 글로벌 명령(글로벌 태그 < Global> < /Global> 또는 우선(priority) 레벨 태그를 갖는 CML에 의해 정의된 바와 같이; 이것은 스코프 태그의 개념과 동등하다)가 아닌 가장 오래된 명령이다. 최상단 메뉴는 메뉴 항목 아래의 명령이 최초로 비활성화되는 한 활성으로 남는다. 비활성화된 명령은 일련의 명령 캐시에서 캐시된다. 명령에 대해 얻어진 확률(또는 인식된 입력의 기타 신뢰치)이 너무 낮거나 상기 명령이 사용자에게 의해 거부되면, 그 말(utterance)은 인식이 가능할 때까지 가장 최근의 명령 캐시에 대해 다시 디코딩된다. 최근에는, 인식된 음성에 대한 적절한 신뢰치를 개발하고자 하는 시도가 이루어져 왔다. 예를 들면, 1996년, 4월 29일-5월 1일 NIST 및 DARPA에 의해 개최된 MITAGS, MD의 LVCSR Hub5 Workshop"에서, 각각의 워드에 신뢰 수준을 덧붙이고자 하는 다른 방법이 제안되었다. 한가지 방법은 워드 의존 특성(말의 훈련량, 최소 및 평균 3 음절(triphone) 발생, 언어 모델 트레이닝에서의 발생, 음운(phonemes/lefemes)의 수, 존속 기간, 음향 스코어(acoustic score)(신속한 정합 및 세부 정합), 음성 비음성(speech non-speech), 문장 의존 특성(신호대 잡음 비, 말하는 속도의 평가, 초당 단어 또는 음운 또는 모음의 수, 언어 모델에 의해 제공된 문장 가능성, 언어 모델의 3 자명(trigram 발생), 문맥내 워드 특성(언어 모델에서의 3자명 발생) 및 화자 프로파일 특성(엑센트, 사투리, 성별, 연령, 말하는 속도, 아이덴티티, 음질, SNR 등) 상에서 훈련된 결정 트리를 이용한다. 여러 확률은 트리의 각 잎(leaves)에 대해 트레이닝 데이터로 계산된다. 이러한 트리를 구축하는 알고리즘은 1993년 Breiman 등의 "Classification and regression trees" Chapman & Hal, 1993에 개시되어 있다. 인식시, 이들 특성 모두 또는 일부는 인식하는 동안 측정되며 각각의 워드에 대해 결정 트리는 신뢰 수준을 제공하는 잎(leave)으로 진행한다. C. Neti, S.Roukos and E.Eide entitled "Word based confidence measures as a guide for stack search in speech recognition" ICASSP97, Munich, Germany, April, 1997에는 IBM 스택 디코더에 의해 리턴된 스코어(score)에 전적으로 의존하는 방법이 개시되어 있다. LVCSR에서, 선형 복귀를 통한 예고자(predictor)를 이용하여 신뢰 수준을 평가하는 다른 방법이 행해진다. 상기 예고자는 워드 존속 기간, 언어 모델 스코어, 프레임당 평균 음향 스코어(최고 스코어) 및 최우선 선택으로 동일한 워드를 갖는 NBEST 목록 부분이다. 본 실시예는 두 가지 방법(결정 트리 및 선형 예고자를 통하여 측정된 신뢰 수준)의 조합을 제공하여, 음향 인식에 제한되지 않고 임의의 번역 과정에서 신뢰 수준을 통계적으로 추출한다.

리와인드(rewind) 또는 리로드(reload) 명령은 사용자에게 페이지 상에서 만나는 모든 명령을 판독하도록 실시될 수 있다. 이것은 동일한 동적 비활성화 원리를 따른다. 특히, 전체 음성 네비게이션 인터페이스가 브라우저에 의해 제공되어 전술한 형태 및 메뉴를 통하여 네비게이션이 가능하게 된다. IBM의 ViaVoice VoiceCenter(이것은 윈도우에 대한 명령 및 제어를 수행한다)와 같은 명령 및 제어 인터페이스에 의해 구현될 수 있다.

능력 기반 프레임(Capability-based frames): 프레임 구성과 마찬가지로, CML 디벨로퍼는 능력[HTML의 cfr< frame> and < /frame> 태그]의 상이한 레벨에 의거하여 다수의 다이얼로그를 제공할 수 있다.

최소 요건 형식(Forms with minimum requirement): 물론 한정된 세트의 가능한 엔트리(한정된 어휘 및 FSG, 선택된 메뉴)를 갖는 형식(또는 메뉴 또는 다른 성분들)만이 삽입된 엔진과 함께 사용될 수 있다. 형식(form)을 갖는 능력을 유지하기 위하여, 브라우저는 적어도 2 개의 상이한 양식을 제공한다. 즉, i) 사전에 규정된 문법(서버 상에서 행해진 편집) 및 어휘 목록으로 채워진 형식: < Global> 명령을 제외한 다른 명령은 비활성으로 된다, ii) 음성을 캡처하여 특별한 특성을 계산하고 이들을 URL에 의해 지시된 서버로 전송하는 애플릿(applet), iii) 원음을 서버로 송신 또는 수신하는 애플릿. 형식 채움 방법의 선택은 서버 또는 브라우저에 의해 행해진다. 이것은 로드될 각각의 문법/어휘 목록이 엔진 요건의 설명을 또한 포함해야 할 것을 요구한다. (ii 및 iii은 YO999-113P에 개시한 서버/통신 능력을 필요로 한다.)

주어진 FSG, 트랜스코딩 및 다수 페이지 또는 프레임 방법에 대한 엔진의 최소 요건을 대략적으로 결정하는데 불이 필요할 것이다.

#### 다중 모델 브라우저(MULTI-MODEL BROWSER)

다중 모델 브라우저를 이용하여, HTML 및 CML이 동일 페이지 상에 제공되거나 또는 개별적으로 전송되거나 동기화될 수 있다. 태그(tag)는 시각 브라우저(< GUI>) 상에 나타나는 것과, 멀티 모델 브라우저(< M-M>) 상에 나타나는 것을 구별하며, GUI 브라우저는 멀티 모델 브라우저가 선택적으로 몇몇 항목들을 표시할 수 있는 동안 모든 것을 표시한다.

이것은 다음의 예에 예시되어 있다.

< GUI>

.....

< /GUI>

< Speech+GUI+M-M>

.....

< /Speech+GUI+M-M>

< GUI+M-M>

< img = ...>

< GUI+M-M>

< Speech>

.....

< GUI>

.....

< /GUI>

< M-M>

.....

< /M-M>

< M-M+Speech>

.....

< /M - M+Speech>

즉, CML 및 HTML은 보존된다. 몇몇 동적 HTML 특성들 및 기능들과 조합 규칙이 부가된다.

부가적인 태그는 동적 ML(결합된 HTML 및 CML)을 제공한다. 예를 들면, TTS가 텍스트를 판독할 때, 그 항목이 색을 바꾸거나 로컬 배경이 색을 바꾼다. 링크의 색 변화는 음성에 의해 인식된 텍스트의 선택과 같이 음성 등에 의해 선택되었다(색 변화는 선택되어 수정될 수 있다).

요약하면, 대화식 브라우저의 도입으로 인터넷 및 전화(IVR)가 통합되어 비용이 절감되고, 그러한 어플리케이션의 적용범위 및 유연성이 확장될 수 있다. 상기 구조는 동일한 액세스 프로토콜 및 정보 구성(HTML, XML, CML 및/또는 WML과 같은 다른 ML)을 이용한다. 본 발명은 많은 상이한 비즈니스 솔루션에 적용될 수 있다. 그 주요한 이점은 높은 유연성을 갖는 독립적인 플랫폼을 구현하기가 쉽고, 중심적으로 전개되어 관리된 기존의 인프라 구조를 이용하기 쉬우며, 높은 안정성과, 낮은 관리 비용을 제공하며, 쉽게 확장 가능/스케일 가능하며, 이 모든 것은 어떠한 인터넷 솔루션에도 해당한다. 양호하게는 어플리케이션 공급자가 쉽게 관리하는 단 하나의 HW 플랫폼만 지원하면 된다. 내용 제공자는, 쉬운 관리를 제공하는 다수의 상이한 클라이언트(순수한 음성, 순수한 GUI 및 음성/GUI의 결합)를 서비스하기 위해 동일 WWW 서버로부터 다양한 어플리케이션을 관리할 수 있다. 또한, 인터넷 기술은 네트워크 상의 모든 서버의 자동 업데이트를 가능하게 한다. 액세스 권한은 또한 인터넷을 통해 액세스할 수 있는 어떠한 장소로부터도 중심적으로 관리될 수 있다. 높은 수준의 보안성이 유지될 수 있다. 현재의 구조는 GSM 단말과 같은 정규 전화기 외에 다른 클라이언트에게 확장될 수 있다. 이것은 퍼스널 बैं킹과 같은 서비스를 관리하는데 요구되는 투자의 측면에서 아주 매력적이다.

또한, 단지 HTTP 프로토콜이 사용될 때, 브라우저(대화식/GUI)와 WWW 서버 사이의 네트워크 상의 트래픽이 최소화될 때, 상기 네트워크는 하나의 일괄적으로(batch) 완전한 응답을 전송하는데 이용된다. 예를 들면, 계좌 번호 및 SSN을 입력하는 동안, 브라우저/대화식 브라우저만이 활성화되고, 그 네트워크는 유향 상태로 된다. 상기 브라우저는 서버가 질의받는 동안 많은 에러를 캐치할 정도로 지능적이다. 에러 정정 및 로컬 명확성(disambiguation)은 예를 들어 CVM 서비스 또는 다운로드된 애플릿을 통해 행해질 수 있다. WWW 서버에 대한 정보는 두 경우에 동일한 프로토콜, 즉, HTTP 프로토콜을 이용하여 전송된다. 응답 정보는 항상 동일한 프로토콜을 이용하여 재전송되며, 그것을 올바른 형식으로 사용자에게 전달하는 것은 브라우저의 일이다. 상기 구조는 이들 어플리케이션의 설계자가 애플릿, 즉, 입력 및 출력 정보를 처리하도록 브라우저에서 실행되는 소형의 프로그램(자바 또는 기타 대화식 절차, 특히, 브라우저가 CVM 상에 기록되는 경우)을 이용할 수 있도록 한다. 이것은 어플리케이션의 유연성을 보다 증가시키며 네트워크의 부담을 보다 감소시킨다. 대화식 브라우저에 있어서, 통상적인 텍스트-음성(Text to Speech) 애플릿이 전송되고 그 다음에 요구된 대역폭이 그래픽 대역폭과 동일하거나 더 작을 것이라는 것을 쉽게 상상할 수 있다. 특히, 이것은 다른 언어에서의 프롬프트들을 합성하도록 행해질 수 있으며, 다른 언어에서의 입력을 인식하도록 행해질 수 있다.

본 발명은 현재의 전화 어플리케이션, 통상적으로는 IVR 어플리케이션에 이러한 이점들을 제공한다. 요금 청구 목적을 위해서는 전화 번호가 이상적이다. 인터넷의 지연은 IVR에서 어떠한 문제도 취하지 않는다. 어떠한 경우에도, Qos(quality of service) 및 RSVP(resource reservation protocol)와 같은 서비스에 의해 지연이 감소될 수 있다. 인터넷 상에서 다른 서버(IVR)와의 접속 능력은 그 능력을 더욱 크게 한다. 서버는 컴퓨터 네트워크 상의 어디에도 위치할 수 있으므로, 필요한 음성 인식 시스템에 대한 액세스는 네트워크 상의 어디에서도 가능하다. TCP/IP 프로토콜을 갖는 패킷 스위칭 기술은 정규 전화 접속보다 양호한 모든 자원을 이용한다. 요금은 사용된 채널의 품질에 의존할 수도 있다. 스위칭된 패킷 네트워크는 보다 저렴하며, 더 적은 데이터를 전송하며, 따라서 더 작은 대역폭을 요구한다. 그 결과 더 낮은 비용의 서비스를 제공한다. 이 개념은 누구에게도 서버를 설정할 기회를 제공하며 정보를 네트워크상에 제공할 기회를 제공한다.

비록 본 명세서에서는 첨부된 도면을 참조하여 예시적인 실시예를 설명하였지만, 본 발명에 따른 시스템 및 방법은 그들 실시예로 한정되는 것이 아니라, 본 발명의 사상 및 범주를 벗어나지 않는 범위 내에서 당업자에 의해 다양하게 변경 및 변형될 수 있다. 그러한 모든 변경 및 변형은 첨부된 청구 범위에 의해서 규정되는 본 발명의 사상 범주 내에 포함된다.

#### (57) 청구의 범위

##### 청구항 1.

대화식 브라우징 시스템에 있어서,

음성 명령을 포함하는 입력 명령을 네비게이션 요구로 변환하는 명령 및 제어 인터페이스와, 사용자에게 표시하기 위한 대화식 사용자 인터페이스를 나타내는 메타 정보를 포함하는 CML(대화식 마크업 언어) 파일을 분석 및 해석하는 CML 프로세서를 구비한 대화식 브라우저와;

상기 명령 및 제어 인터페이스에 의해 해석하기 위한 입력 명령을 디코딩하고, 합성 오디오 출력을 발생하기 위해 상기 CML 프로세서에 의해 제공되는 메타 정보를 디코딩하는 대화식 엔진과;

상기 네비게이션 요구를 내용 서버로 송신하고 그 네비게이션 요구에 기초하여 상기 내용 서버로부터 CML 파일을 수신하는 통신 스택

을 포함하는 대화식 브라우징 시스템.

##### 청구항 2.

제1항에 있어서, 상기 입력 명령은 다중 양식 입력을 포함하는 것인 대화식 브라우징 시스템.

##### 청구항 3.

제1항에 있어서, 대화식 가상 머신(CVM)을 더 포함하고,

상기 대화식 브라우저는 상기 CVM의 상부에서 실행하고 상기 CVM을 통해 상기 대화식 엔진으로 대화식 서비스의 요구를 생성하는 것인 대화식 브라우징 시스템.

##### 청구항 4.

제3항에 있어서, 상기 대화식 브라우징 시스템은 클라이언트 장치에서 실행하고, 상기 통신 스택은 내용 서버에 상기 클라이언트 장치의 대화 자격을 등록하는 대화식 프로토콜을 포함하는 것인 대화식 브라우징 시스템.

##### 청구항 5.

제3항에 있어서, 상기 대화식 브라우저, CVM 및 대화식 엔진 중 하나는 네트워크 상에 분배되는 것인 대화식 브라우징 시스템.

##### 청구항 6.

제1항에 있어서, 상기 통신 스택은 상기 네비게이션 요구를 송신하고 상기 CML 파일을 수신하는 표준 네트워킹 프로토콜을 수행하는 것인 대화식 브라우징 시스템.

##### 청구항 7.



제6항에 있어서, 상기 표준 네트워킹 프로토콜은 TCP/IP(송신 제어 프로토콜/인터넷 프로토콜), HTTP(하이퍼텍스트 송신 프로토콜), WAP(무선 어플리케이션 프로토콜), VoIP(인터넷을 통한 음성 프로토콜) 및 이들의 조합 중 하나를 포함하는 것인 대화식 브라우징 시스템.

청구항 8.

제1항에 있어서, 내용 서버의 유산 정보 포맷을 CML 파일로 변환하는 트랜스코더를 더 포함하는 것인 대화식 브라우징 시스템.

청구항 9.

제8항에 있어서, 상기 트랜스코더는 상기 대화식 브라우저를 통해 사용자와의 다이얼로그를 발생하도록 어플리케이션의 논리 정보를 액세스 및 해석하는 것인 대화식 브라우징 시스템.

청구항 10.

제8항에 있어서, 상기 트랜스코더는 대화식 브라우저, 대화식 프록시 서버, 내용 서버 중 하나에서 실행되거나, 이들의 조합 사이에 분배되는 것인 대화식 브라우징 시스템.

청구항 11.

제8항에 있어서, 상기 트랜스코더는 대화식 브라우징 시스템이 실행하는 머신의 대화 자격에 기초하여 주문 CML 파일을 발생하는 것인 대화식 브라우징 시스템.

청구항 12.

제1항에 있어서, 상기 CML 파일은, (1) 다른 CML 구성 요소를 그룹화한 페이지 구성 요소; (2) 상기 대화식 브라우저에 의해 구두로 전달된 출력을 지정하는 본문 구성 요소; (3) 사용자에게 구두로 전달되는 서두 텍스트 및 프롬프트 텍스트를 각각 갖고 대응하는 선택이 선택될 때 CML 구성 요소를 식별하는 목표 어드레스와 관련되는 선택의 목록을 포함하는 메뉴 구성 요소; (4) 적어도 하나의 정보 아이템 및 이 적어도 하나의 정보 아이템을 전송하기 위한 목표 어드레스를 입력하는 형식 구성 요소; 및 (5) 이들의 조합 중 하나를 포함하는 것인 대화식 브라우징 시스템.

청구항 13.

제12항에 있어서, 상기 목표 어드레스는 URL(uniform resource locator) 및 소켓 어드레스 중 하나를 포함하는 것인 대화식 브라우징 시스템.

청구항 14.

대화식 브라우징을 제공하는 방법에 있어서,

음성 명령을 포함하는 입력 명령을 수신하는 단계와;

상기 입력 명령을 복수의 대화식 엔진의 적어도 하나에 의해 디코딩하는 단계와;

상기 디코딩된 입력 명령에 기초하여, CML(대화식 마크업 언어) 파일이 사용자에게 표시하기 위한 대화식 사용자 인터페이스를 나타내는 메타 정보를 포함하는 CML을 검색하는 네비게이션 요구를 발생하는 단계와;

상기 네비게이션 요구를 송신하고 표준 네트워킹 프로토콜을 사용하여 요구된 상기 CML 파일을 액세스하는 단계와;  
상기 CML 파일을 포함하는 메타 정보를 분석 및 해석하여 상기 CML 파일의 정보 내용의 오디오 표시를 제공하는 단계

를 포함하는 대화식 브라우징 제공 방법.

청구항 15.

제14항에 있어서, 상기 CML 파일이 액세스되는 엔티티에 대화 자격을 등록하는 단계와;

상기 등록된 자격에 기초하여 CML 파일을 주문하는 단계를 더 포함하는 것인 대화식 브라우징 제공 방법.

청구항 16.

제14항에 있어서, 유산 정보 포맷을 포함하는 파일에 액세스하는 네비게이션 요구를 발생하는 단계와;

상기 유산 정보 포맷을 CML 파일로 변환하는 단계를 더 포함하는 것인 대화식 브라우징 제공 방법.

청구항 17.

제16항에 있어서, 상기 변환 단계는 파일이 액세스되는 내용 서버와 관련된 프록시를 트랜스코딩함으로써 실행되는 것인 대화식 브라우징 제공 방법.

청구항 18.

제14항에 있어서, 상기 CML 파일과 관련된 어플리케이션의 논리 정보를 액세스 및 해석하여 다이얼로그를 발생하는 단계를 더 포함하는 것인 대화식 브라우징 제공 방법.

청구항 19.

대화식 브라우징을 제공하는 방법을 실행하도록 머신에 의해 실행 가능한 명령 프로그램을 명확하게 구현하는 머신에 의해 판독 가능한 프로그램 기억 장치로서, 상기 대화식 브라우징 제공 방법은,

음성 명령을 포함하는 입력 명령을 수신하는 단계와;

상기 입력 명령을 복수의 대화식 엔진의 적어도 하나에 의해 디코딩하는 단계와;

상기 디코딩된 입력 명령에 기초하여, CML(대화식 마크업 언어) 파일이 사용자에게 표시하기 위한 대화식 사용자 인터페이스를 나타내는 메타 정보를 포함하는 CML을 검색하는 네비게이션 요구를 발생하는 단계와;

상기 네비게이션 요구를 송신하고 표준 네트워킹 프로토콜을 사용하여 요구된 상기 CML 파일을 액세스하는 단계와;

상기 CML 파일을 포함하는 메타 정보를 분석 및 해석하여 상기 CML 파일의 정보 내용의 오디오 표시를 제공하는 단계를 포함하는 프로그램 기억 장치.

청구항 20.

제19항에 있어서, 등록된 자격에 기초하여 엔티티가 상기 CML 파일을 주문할 수 있도록 상기 CML 파일이 액세스되는 엔티티에 대화 자격을 등록하는 단계를 실행하는 명령을 더 포함하는 것인 프로그램 기억 장치.

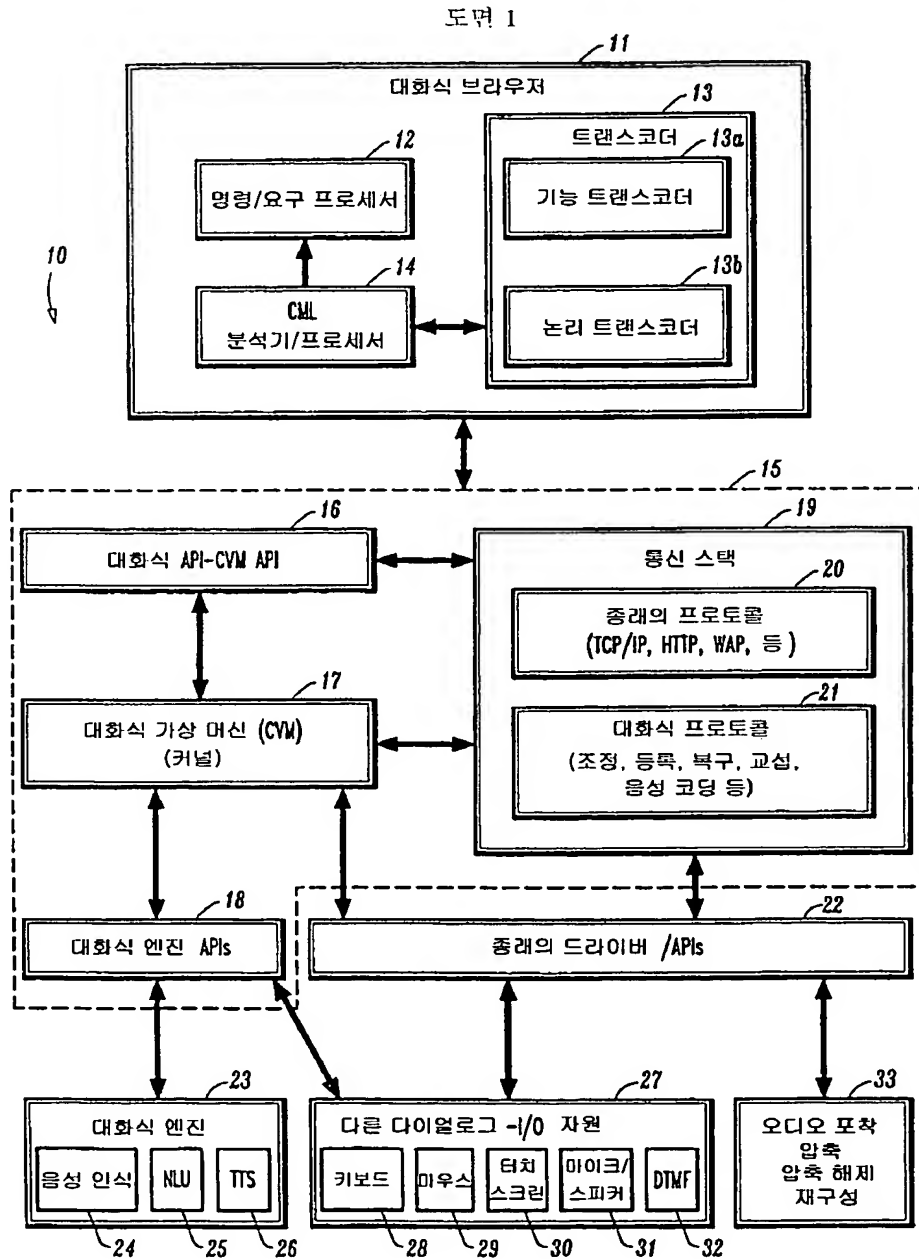
청구항 21.

제19항에 있어서, 유산 정보 포맷을 포함하는 파일에 액세스하는 네비게이션 요구를 발생하는 단계와, 상기 유산 정보 포맷을 CML 파일로 변환하는 단계를 실행하는 명령을 더 포함하는 것인 프로그램 기억 장치.

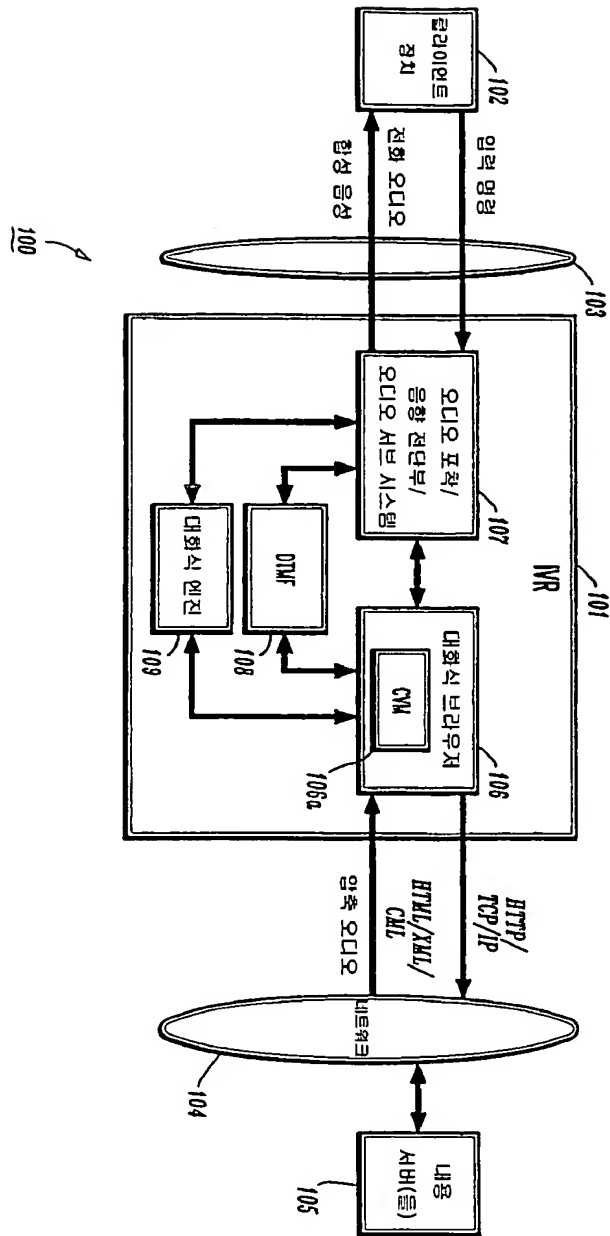
청구항 22.

제19항에 있어서, 상기 CML 파일과 관련된 어플리케이션의 논리 정보를 액세스 및 해석하여 다이얼로그를 발생하는 단계를 실행하는 명령을 더 포함하는 것인 프로그램 기억 장치.

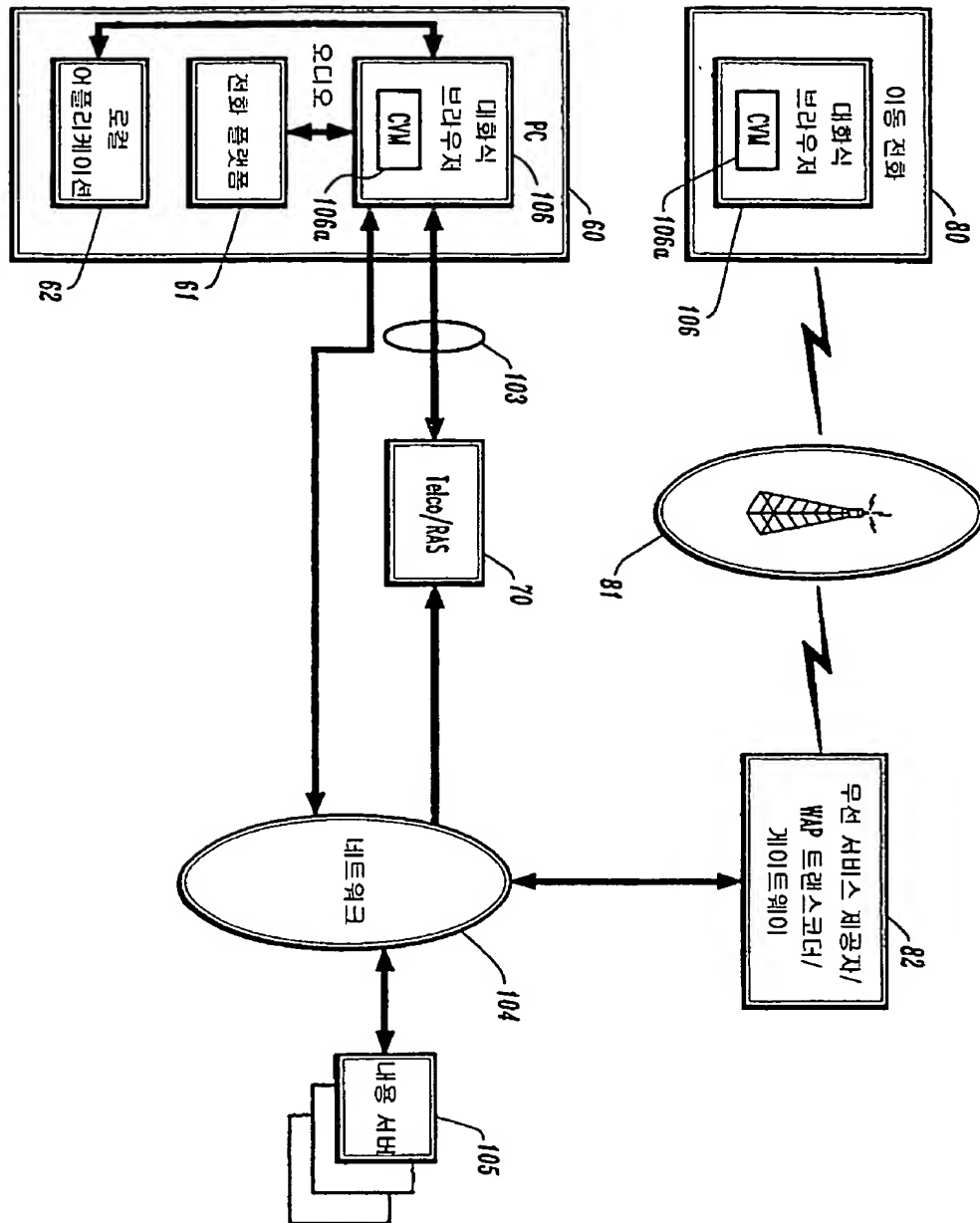
도면



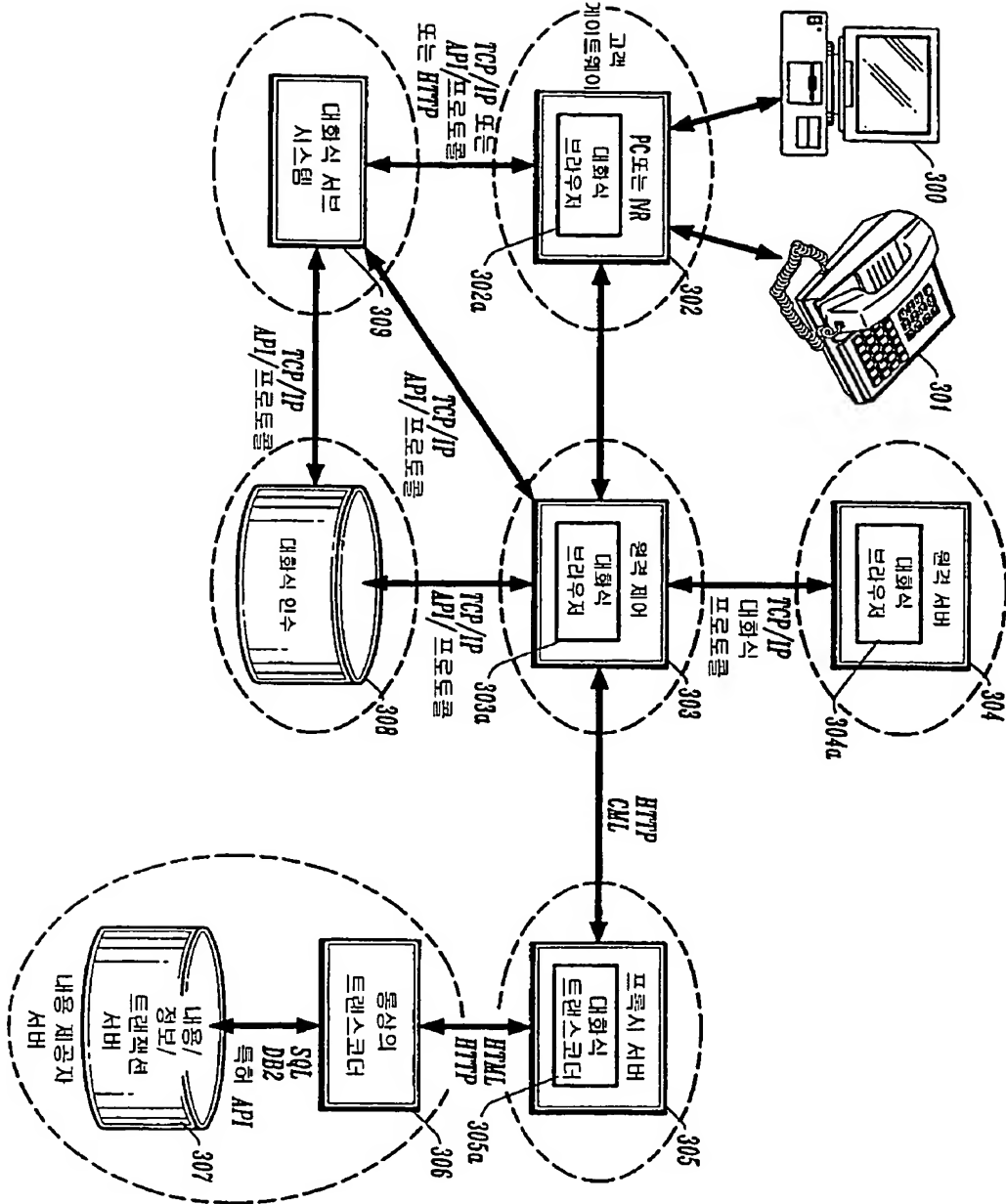
도면 2



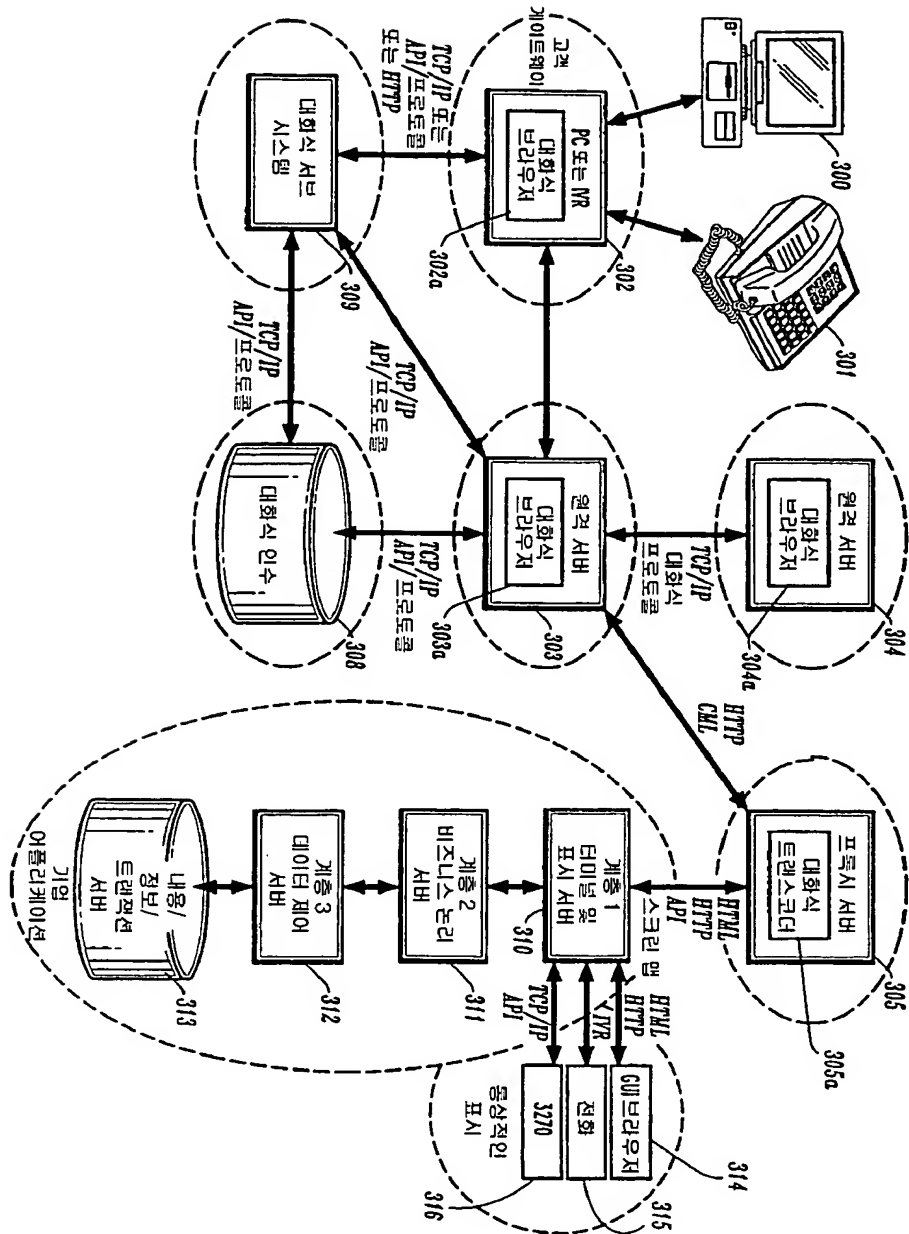
도면 3



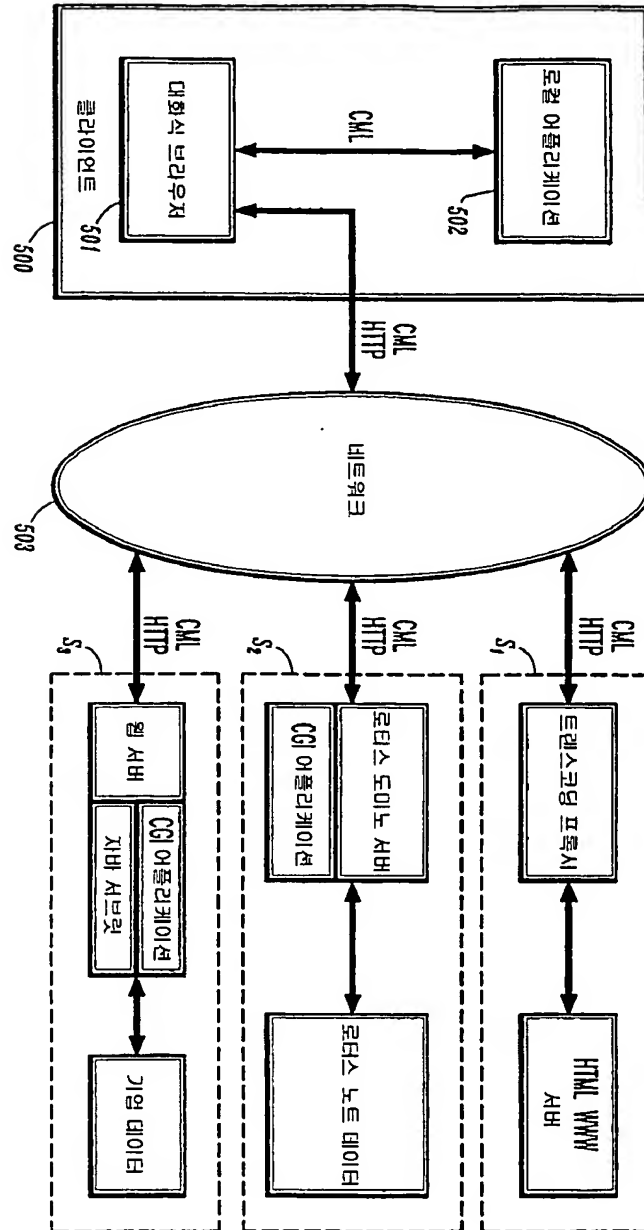
도면 4a



도면 4b

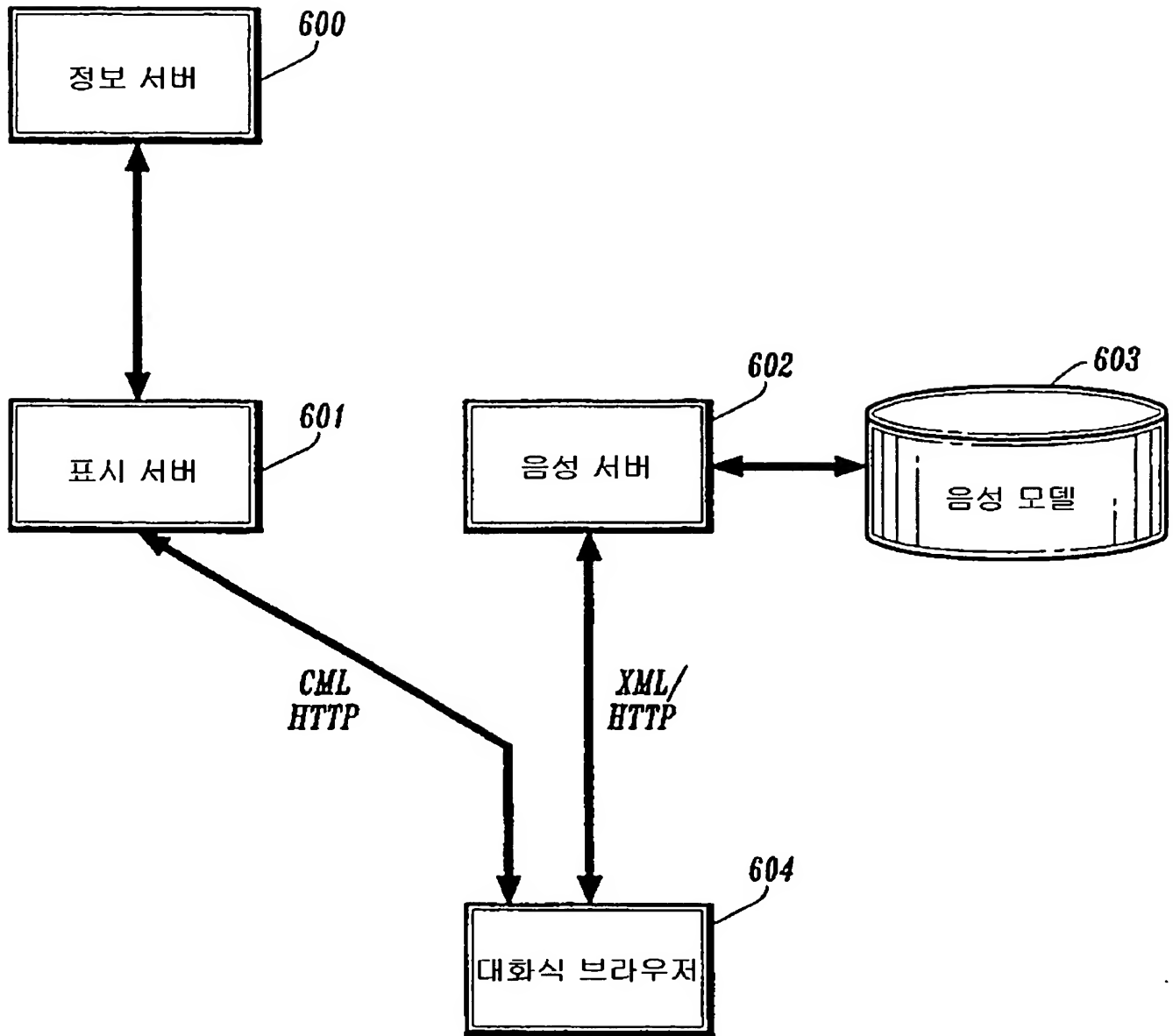


도면 5

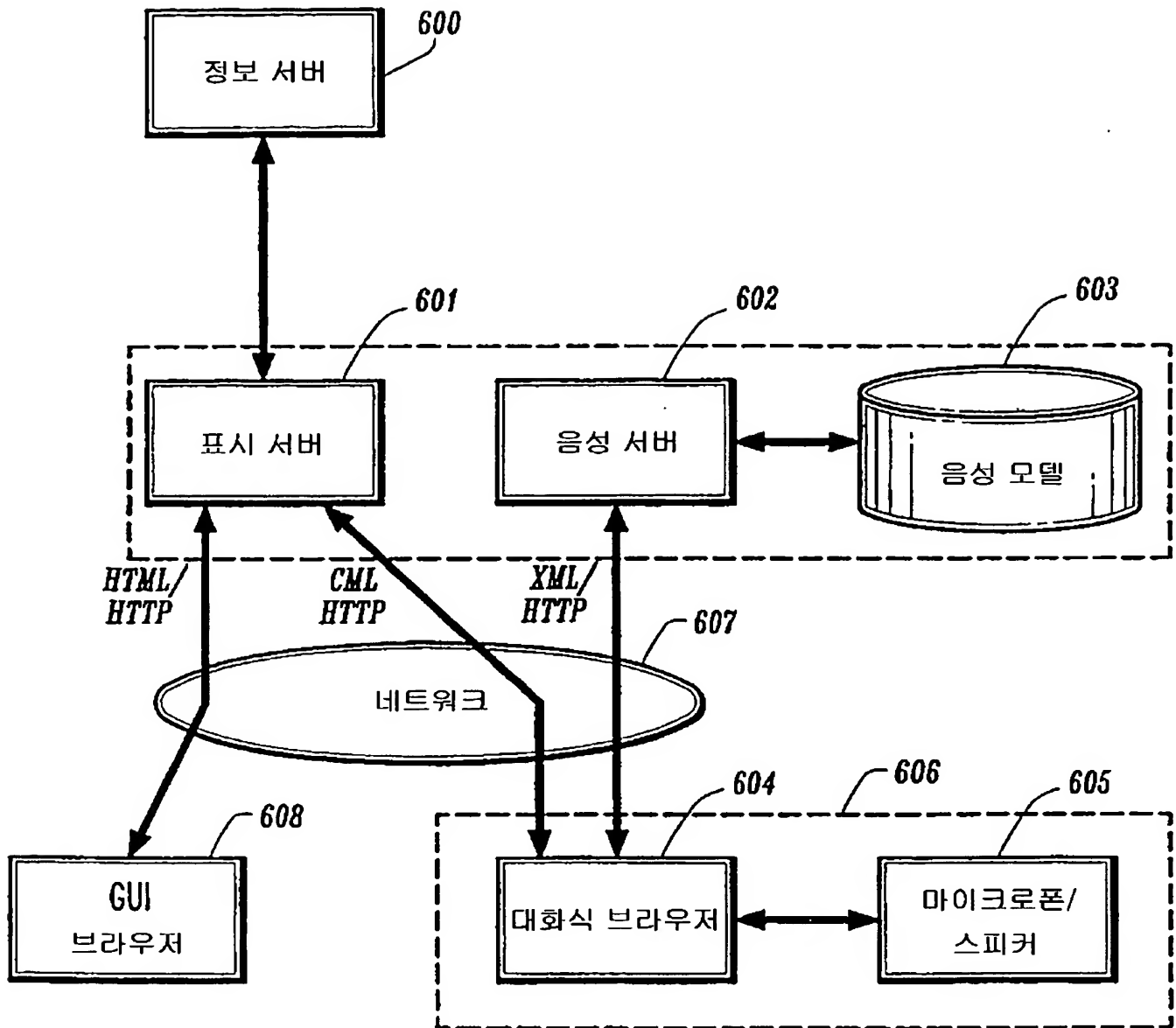




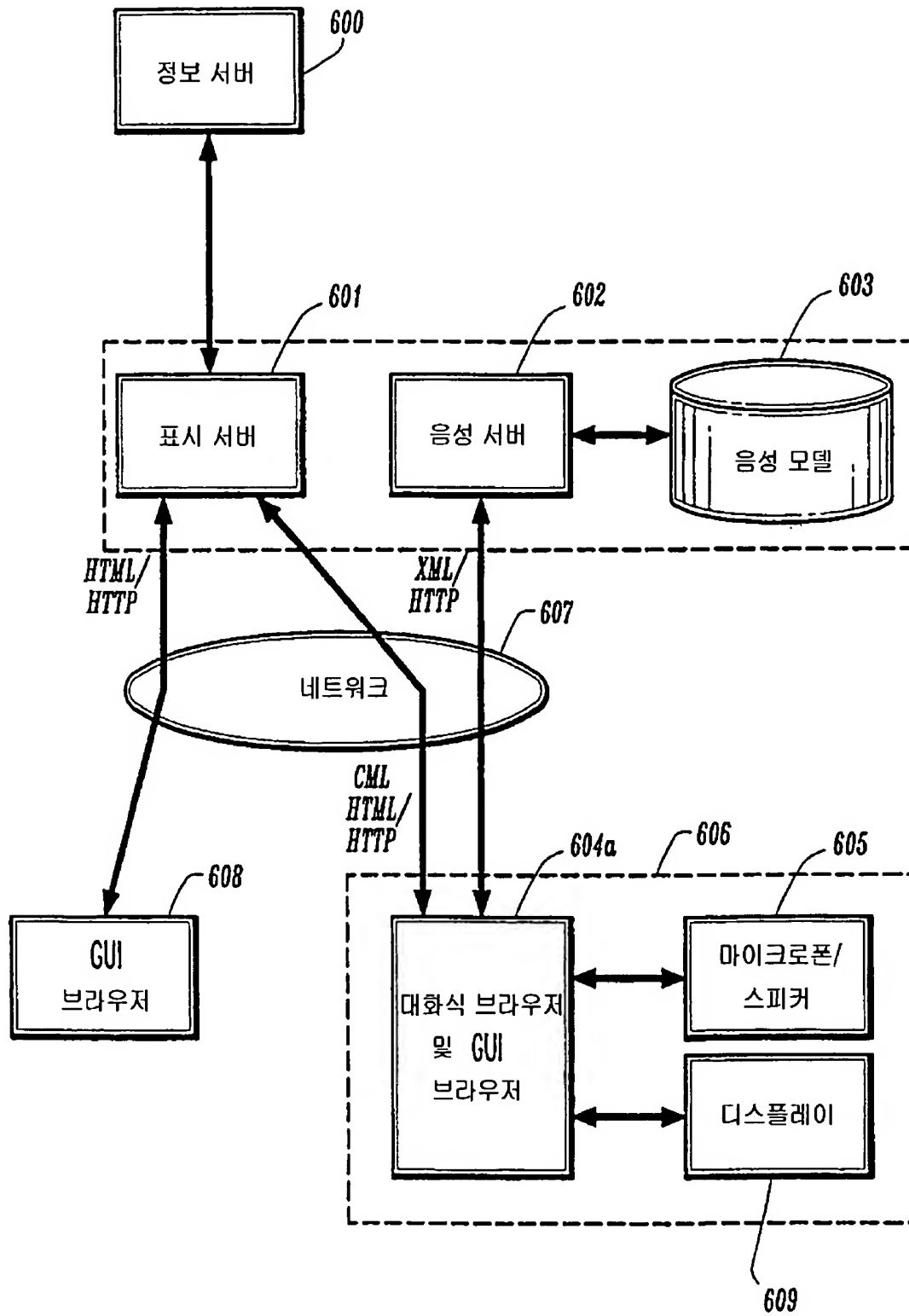
도면 6



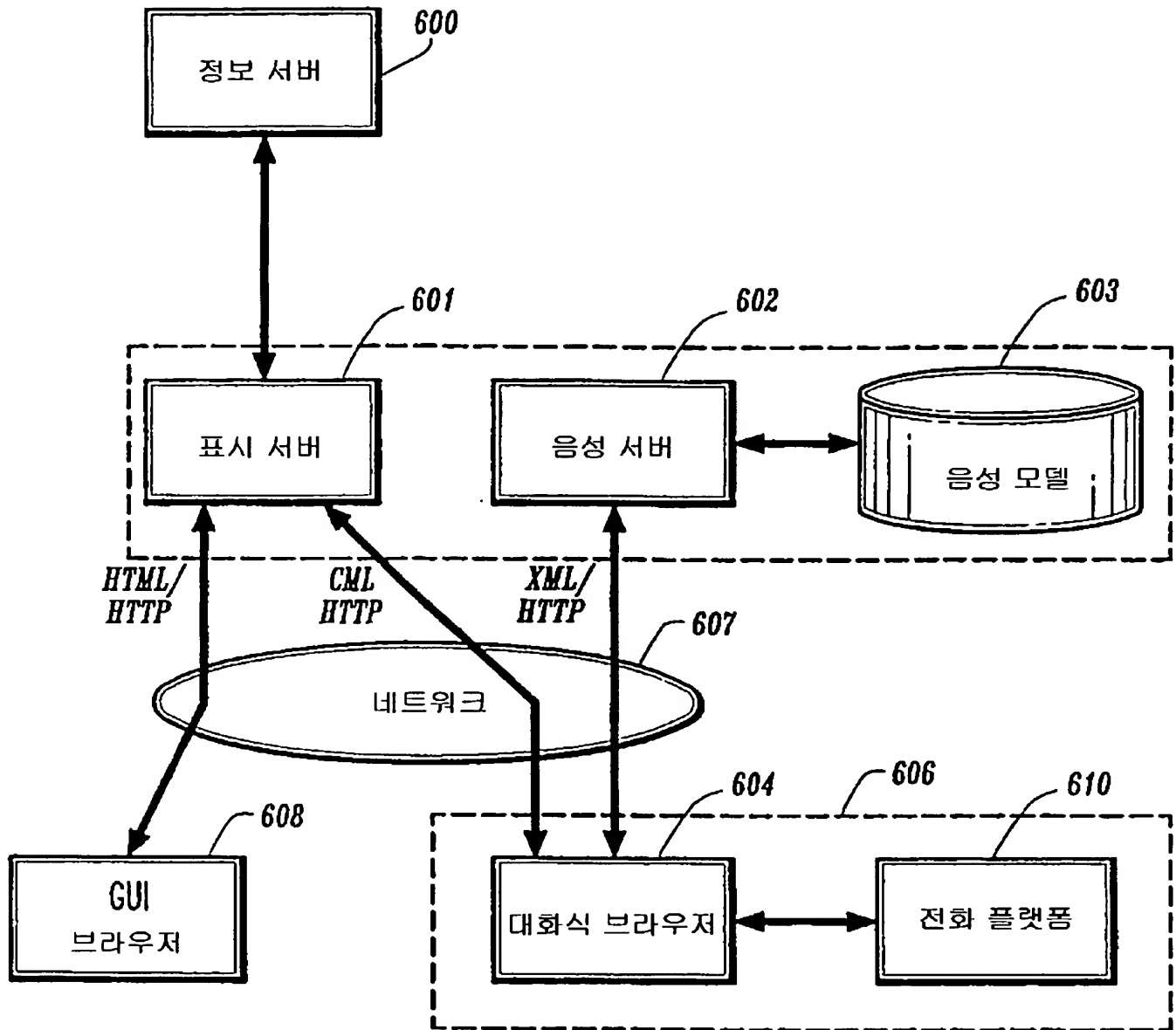
도면 7



도면 8



도면 9



도면 10

